

AD-A168 517

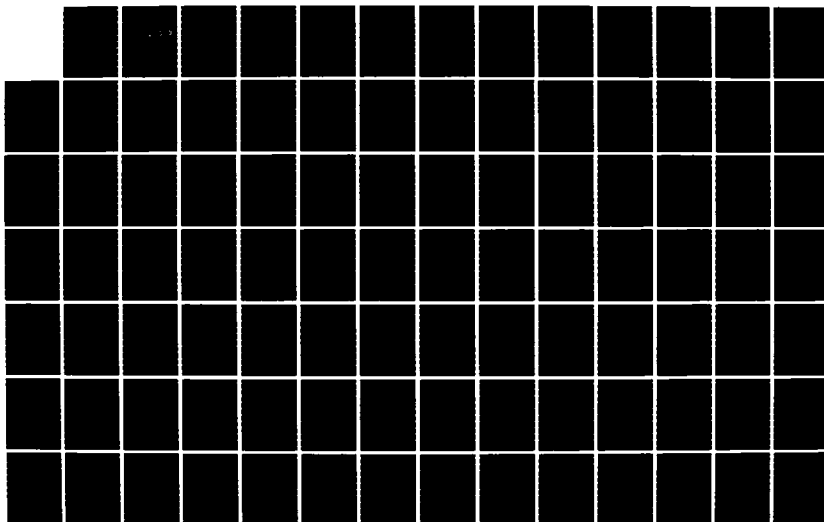
DEVELOPMENT OF AN AUTOMATED MICROCOMPUTER
KNOWLEDGE-BASED INTEGRATED CON. (U) NAVAL POSTGRADUATE
SCHOOL MONTEREY CA R L BEARD MAR 86

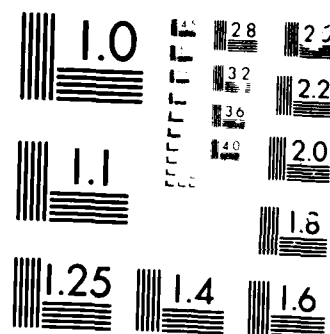
1/3

UNCLASSIFIED

F/8 9/2

NL





MICROFILM

2

AD-A168 517

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC
ELECTE
JUN 20 1986
S D

THESIS

MMC FILE COPY

DEVELOPMENT OF AN AUTOMATED MICRO-COMPUTER
KNOWLEDGE-BASED INTEGRATED CONFIGURATION
MANAGEMENT SYSTEM FOR THE STOCK POINT
LOGISTICS INTEGRATED COMMUNICATIONS
ENVIRONMENT (SPLICE) PROJECT MANAGEMENT STAFF

by

Robert Lee Beard III

March 1986

Thesis Advisor:
Co-Advisor:

Norman R. Lyons
Barry A. Frew

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION			1b RESTRICTIVE MARKINGS			
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited			
2b DECLASSIFICATION/DOWNGRADING SCHEDULE						
4 PERFORMING ORGANIZATION REPORT NUMBER(S)			5 MONITORING ORGANIZATION REPORT NUMBER(S)			
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b OFFICE SYMBOL (if applicable) 54		7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School		
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000			7b ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000			
8a NAME OF FUNDING/SPONSORING ORGANIZATION		8b OFFICE SYMBOL (if applicable)		9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c ADDRESS (City, State, and ZIP Code)			10 SOURCE OF FUNDING NUMBERS			
			PROGRAM ELEMENT NO		PROJECT NO	TASK NO
						WORK UNIT ACCESSION NO
11 TITLE (include Security Classification) DEVELOPMENT OF AN AUTOMATED MICRO-COMPUTER KNOWLEDGE-BASED INTEGRATED CONFIGURATION MANAGEMENT SYSTEM FOR THE STOCK POINT LOGISTICS COMMUNICATIONS ENVIRONMENT (SPLICE) PROJECT MANAGEMENT STAFF						
12 PERSONAL AUTHOR(S) Robert L. Beard III						
13a TYPE OF REPORT Master's Thesis		13b TIME COVERED FROM TO		14 DATE OF REPORT (Year, Month, Day) 1986 March		15 PAGE COUNT 425
16 SUPPLEMENTARY NOTATION						
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number) SPLICE, Expert System, Knowledge-Based System, Configuration Management System, DBMS			
FIELD	GROUP	SUB-GROUP				
19 ABSTRACT (Continue on reverse if necessary and identify by block number) This thesis documents the development of a micro-computer knowledge-based integrated configuration management system for use by Naval Supply Systems Command (NAVSUP) Stock Point Logistics Integrated Communications Environment (SPLICE) Project Staff. A myriad of configuration heuristics associated with the configuration of a SPLICE site are identified. It also provides SPLICE project staff personnel a more accurate, reliable and efficient method of performing the configuration process and managing the overall project. The development of this integrated configuration management system employs both a prototype and software engineering methodology. The integrated configuration management system will be developed using custom generated software and the logical integration of several off-the-shelf commercial software packages.						
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS				21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a NAME OF RESPONSIBLE INDIVIDUAL Norman R. Lyons				22b TELEPHONE (Include Area Code) (408) 646-2666		22c OFFICE SYMBOL 54Lb

Approved for public release; distribution is unlimited.

Development of an Automated Micro-computer Knowledge-based
Integrated Configuration Management System for the Stock
Point Logistics Integrated Communications Environment
(SPLICE) Project Management Staff

by

Robert L. Beard III
Lieutenant Commander, Supply Corps, United States Navy
BE CS, University of New Mexico, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1986

Author:

Robert L. Beard III
Robert L. Beard III

Approved by:

Norman R. Lyons
Norman R. Lyons, Thesis Advisor

Barry A. Frew
Barry A. Frew, Co-Thesis Advisor

Willis R. Greer, Jr.
Willis R. Greer, Jr., Chairman,
Department of Administrative Sciences

Kneale T. Marshall
Kneale T. Marshall
Dean of Information and Policy Sciences

ABSTRACT

This thesis documents the development of a micro-computer knowledge-based integrated configuration management system for use by Naval Supply Systems Command (NAVSUP) Stock Point Logistics Integrated Communications Environment (SPLICE) Project Staff. A myriad of configuration heuristics associated with the configuration of a SPLICE site are identified. It also provides SPLICE project staff personnel a more accurate, reliable and efficient method of performing the configuration process and managing the overall project.

The development of this integrated configuration management system employs both a prototype and software engineering methodology. The integrated configuration management system will be developed using custom generated software and the logical integration of several off-the-shelf commercial software packages.

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	



THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

I.	INTRODUCTION -----	7
A.	PURPOSE -----	7
B.	BACKGROUND -----	8
C.	SCOPE -----	9
II.	CONFIGURATION RULES -----	12
A.	BASIC CONFIGURATION RULES -----	12
B.	UNIQUE CONFIGURATION RULES -----	16
1.	Hardware -----	16
2.	Software -----	18
3.	Manuals and Documentation -----	19
4.	Training -----	20
5.	Maintenance -----	20
6.	Other -----	21
7.	Discount and Escalation Rates -----	21
III.	METHODOLOGY USED TO DEVELOP THE SYSTEM -----	23
A.	PROTOTYPE -----	25
B.	SOFTWARE ENGINEERING METHODOLOGY -----	27
1.	Planning -----	28
2.	Development -----	32
3.	Maintenance -----	34
C.	SUMMARY -----	37
IV.	SYSTEM EXECUTION DIALOGUE -----	39
A.	SYSTEM INITIATION -----	40
B.	CONFIGURE A SITE -----	40

C.	PERFORM FINANCIAL ANALYSIS ON SITE DATA -----	41
D.	INTERACT WITH THE CONFIGURATION MANAGEMENT AND REPORT GENERATION SUB-SYSTEM -----	42
1.	Load New Delivery Order Data -----	42
2.	Load Serial Number and Manual Data -----	43
3.	Generate a Maintenance Delivery Order -----	45
4.	Generate a Report -----	46
E.	REVIEW THE ON-LINE USER'S MANUAL -----	46
F.	TERMINATE SYSTEM EXECUTION -----	47
V.	COST BENEFIT AND EFFECTIVENESS -----	48
	LIST OF REFERENCES -----	53
	APPENDIX A: USER'S MANUAL -----	54
	APPENDIX B: MAINTENANCE MANUAL -----	138
	BIBLIOGRAPHY -----	423
	INITIAL DISTRIBUTION LIST -----	424

I. INTRODUCTION

A. PURPOSE

The Naval Supply Systems Command (NAVSUP) Stock Point Logistics Integrated Communications Environment (SPLICE) Project Manager is tasked with the responsibility to oversee, direct and review all aspects of the SPLICE project. More specific responsibilities of the SPLICE Project Manager include:

1. ADP equipment acquisition
2. ADP software development
3. Coordination of installations and implementations with field activities

In order to perform the latter of the above responsibilities, the Project Manager must maintain a complete history of all configuration components and component changes. This requirement applies to each component of hardware, software and documentation for the complete fifteen year life cycle of the project.[Ref. 1]

This thesis is designed to provide the Project Manager the capability to perform these functions in an automated manner. A micro-computer knowledge-based integrated configuration management system is seen as the means to accomplish the task. To aid in the development of such a system and reduce development time and difficulty,

functional off-the-shelf commercial packages, where feasible, were used. The system was also designed as a user-friendly interactive system.

B. BACKGROUND

In 1977, NAVSUP conceived and developed the SPLICE project to accomplish the following goals:

1. Provide state-of-the-art local and long haul telecommunications capabilities to sixty-two NAVSUP Stock Points
2. Provide interactive and distributed automated data processing (ADP) capabilities to SPLICE sites
3. Provide capacity relief to aging Burroughs hosts at the Stock Points
4. Standardize and upgrade, via mass replacement, the myriad of minicomputers existing at Stock Points

To achieve these goals, NAVSUP initiated a competitive solicitation for "fault-tolerant" hardware and software. The solicitation was completed in November 1983 and the contract was awarded to Federal Data Corporation (FDC). FDC proposed TANDEM hardware and software to meet most of the solicitation processing and local communication requirements. Network System Corporation hardware and software were proposed to meet the local inter-host communication requirements.

Shortly after the SPLICE contract award, hardware and software components had to be ordered. NAVSUP faced a dilemma. Only a few SPLICE personnel had worked closely with the SPLICE acquisition benchmark and negotiations.

These few people were the only personnel that had sufficient knowledge of the system to configure and generate delivery orders. These personnel developed initial orders by hand to meet the immediate need. Numerous minor errors were encountered with initial orders. FDC corrected the orders to the government and received additional compensation for their efforts.

This manual configuration process was later automated using a software product called SUPERCALC2. It has subsequently transitioned to LOTUS 1-2-3. The basic method of developing these orders remained virtually manual. These few SPLICE personnel, with FDC assistance, developed a series of "rules of thumb" used to configure individual site systems. Many of the original SPLICE group have moved on, taking their knowledge of the system with them.

C. SCOPE

A knowledge-based integrated configuration management software system designed to run on a micro-computer was proposed by a former Fleet Material Support Office¹ (FMSO)

¹FMSO is the Central Design Agency for all NAVSUP software development projects. As such, FMSO is responsible for the project development of the SPLICE project under the guidance and direction of the Systems Commander Project Manager, NAVSUP.

SPLICE project officer² to codify these "rules of thumb." The proposed integrated configuration management system will provide NAVSUP with the capability to develop and maintain SPLICE configurations and delivery orders and to perform configuration management for the overall project. The proposed integrated system will be composed of three software modules designed to:

1. Configure initial SPLICE site systems by answering a series of configuration related questions
2. Restructure the system configurer output file into a format compatible for financial and "what-if" analysis
3. Restructure the financial module output file into a format compatible for entry into a data base management system
4. Generate a series of configuration management reports to:
 - a. obtain an overall project report
 - b. obtain a report for a particular site
 - c. obtain a report for a delivery order issued on a particular date
5. Generate a maintenance delivery order for a specific SPLICE site
6. Generate a set of mailing labels for all designated SPLICE sites

²Lieutenant Commander Edward J. CASE, Supply Corps, United States Navy served as SPLICE project officer from September 1981 to August 1984. LCDR CASE was enrolled as a student at the Naval Postgraduate School from October 1984 to March 1986. Much of the research and development of the micro-computer knowledge-based integrated configuration management system is attributed to the prior knowledge, experience and efforts of LCDR CASE.

Development of the micro-computer knowledge-based integrated configuration management system and successful implementation of the configuration heuristics will provide the NAVSUP SPLICE project manager with the capability to perform all assigned configuration management tasks.

II. CONFIGURATION RULES

The success of the knowledge-based integrated configuration management system is largely dependent upon the accurate implementation of the numerous heuristics involved in the configuration of SPLICE site components. Heuristics which must be considered during the configuration process fall into two categories:

1. basic configuration rules which apply to all contract line items under consideration
2. specific configuration rules which apply only to selective contract line items

A breakdown and discussion of these two categories of heuristics is provided below.

A. BASIC CONFIGURATION RULES

A TANDEM processing system consists of a mainframe and its free standing peripherals. A small standard mainframe normally includes two cabinets:

1. processor (CPU) cabinet
2. tape cabinet

The processor cabinet houses the processing units (CPUs) and associated power supplies. The tape cabinet houses a magnetic tape unit, Diagnostic Link control panel, I/O patch panels, battery pack or I/O power supply modules. The I/O patch panels provide attachment points for the signal cables

of various peripherals (ex: CRT terminals. line printers, large capacity disks. etc.). Patch panels are connected to the device controllers residing in the system cabinets through internal cabling.

Additional cabinets (ex: processor, tape, patch panel and expansion) may be added as necessary. Patch panel cabinets provide space for additional patch panels when tape cabinet capacity is inadequate. Generally, mainframe cabinets are fastened together side-by-side to form a single unit.

When two processor cabinets are used in a system and both cabinets contain I/O controllers. additional space for I/O only power supplies may be required. Additional I/O only power supplies may be housed in system expansion cabinets.

System expansion cabinets are required for systems with three or more processor cabinets (or with two processor cabinets connected as noted above). I/O only cabinets must be ordered when system composition reaches four system cabinets. I/O only cabinets may also be necessary to accommodate increased I/O device loads.

Twenty-four I/O slots (four identical backplane assemblies each containing six board slots) are available in a NonStop TXP processor cabinet. The placement of controller boards may result in the need to order additional system or I/O expansion cabinets.

Include one Operations and Service Processor (OSP) with each system.

Every processing unit is supplied with a standard power supply module. The power supply provides several DC voltage levels for use by the CPU, memory and I/O device controllers. No redundant power supply exists for the CPU. Redundancy at the processor unit is obtained with multiple processor units.

In a simple configuration all device controllers are connected to both I/O channels. A simple configuration may be two processors with limited memory and I/O capability.

The I/O channel for a processing unit can accommodate up to thirty-two I/O device controllers. Each device controller can control a maximum of eight devices.

Every I/O controller has two addresses, is dual-ported and is connected to two processor channels.

A one-to-one relationship exists between a controller address and the number of circuit boards it represents with the following exceptions:

1. One 3106 disc controller consists of two boards
2. The 6303 asynchronous controller board accounts for four controller addresses regardless of the number of communications lines it controls. The four controller addresses can represent from one to three boards: one 6303 plus one or two 6304 expansion boards

A fiber optic link (FOX) permits multiple configurations of up to sixteen TANDEM processors each to be directly interfaced. One 6700 FOX controller is required per node.

A special backplane upgrade and replacement is included with the 6700 controller. The FOX controller must reside in the first six (leftmost) I/O slots in the system directly under processor number zero. Any system configuration which includes FOX must consider this requirement. Some such systems may require an additional I/O cabinet to accommodate all controllers. The FOX controller consumes approximately forty-eight amperes of +5 VDC power and may impact the power configuration considerations.

A five strand one-hundred meter air plenum pre-terminated cable, model 7618, should be utilized. The 7618 cable is UL approved for use in air plenum spaces (under raised floors, above false ceilings, etc.) without need for installation in conduit (UL rating VW1). The fifth strand is provided as an integral part of the cable and serves as a spare in case of breakage or intermittent voltage levels.

Terminal communications to the TANDEM hosts is accomplished via specific processor resident ASYNC or SYNC controllers or is off-loaded to a 6100 controller (communications processor).

Network Systems Corporation (NSC) HYPERchannel products enable two or more computer systems to communicate with each other at multi-megabit rates. A HYPERchannel network consists of one or more coaxial cables running the length of the computer room. HYPERchannel adapters are tapped into

the cable and connected to the applicable hosts at designated high speed I/O channel ports. User or NSC software creates the processing sessions among the hosts.

B. UNIQUE CONFIGURATION RULES.

Unique rules must be applied during the configuration process in addition to the basic configuration rules. These additional heuristics apply to all classes of available options (ex: hardware, software, documentation, etc.). The discussions which follow highlight these additional considerations.

1. Hardware

Unique configuration heuristics described below apply to hardware line items.

1. One to four CPUs require one system cabinet and one patch panel. Each CPU is ordered with two megabytes of memory and is augmented with an additional two megabytes of memory.
2. Five to eight CPUs require two system cabinets, one patch panel and one expansion cabinet.
3. Nine to twelve CPUs require three system cabinets, two patch panels and one expansion cabinet.
4. Larger configurations are built using multiples of the above three rules.
5. The FLOATING POINT ARITHMETIC microcode for FORTRAN processing is only ordered for the two FMSO sites (Sites 02 and 03).
6. An Operations and Service Processor (OSP), with a TANDEM 6530 CRT attached, is ordered for each configuration of sixteen processors or portions thereof. The OSP must be capable of using an

attached Centronics Printer with a printer interface unit that permits switching among two OSPs.

7. Each system cabinet requires three I/O power modules.
8. Each system cabinet has twenty-four slots. Each controller (ex: disk controller, LP/CR controller, etc.) occupies two slots.
9. One disk controller is needed for every two disk units ordered.
10. Disk controllers must be ordered in pairs.
11. One disk patch panel is required for every four disk controllers.
12. HYPERchannel adapters may only be ordered by sites designated as stock points. Available HYPERchannel adapters are listed as follows:
 - a. A140 - UNIVAC host interface.
 - b. A150 - Burroughs B4800 host interface. An EBCDIC-to-ASCII Conversion RAM board is ordered with each A150 adapter to facilitate TANDEM-to-Burroughs communications.
 - c. A220 - IBM host interface.
 - d. A400 - Standard minicomputer interface used for TANDEM and PERKIN-ELMER hosts. Each adapter can support up to four CPUs. This is the only adapter which can exceed the one-to-one relationship between processors and adapters.
 - e. A510 - FIPS Standard host interface. HYPERchannel component pricing is based upon the assumption that the maximum number of components to achieve the maximum discount have already been ordered.
13. Each HYPERchannel cabinet will accommodate up to three adapters. If TANDEM and Burroughs machines are greater than fifty feet apart, a HYPERchannel cabinet is needed for each machine. Coaxial cables in lengths from 500 to 5000 feet may be ordered as needed.
14. One patch panel cabinet is required for every ten patch panels (any type).

15. 6100 Communications Subsystem Base units come with a cabinet with room to accommodate fifteen Line Interface units (LIUs) and two Subsystem Base Add-on units. Each Subsystem Base Add-on unit can accommodate an additional fifteen LIUs. Three cable size options are available for connecting the 6100 Subsystem to hosts. Only the 60M option is ordered. Each Subsystem Base unit and Add-on unit requires two cables.
16. One TANDEM HYPERchannel patch panel is required for every four TANDEM HYPERLINK controllers.
17. One tape controller is needed for every tape drive unit.
18. One LP/CR controller is required for every line printer, card reader or card reader punch unit.
19. All TANDEM 6530 CRTs are ordered with the word processing option.
20. One ASYNC patch panel is required for each ASYNC controller. An ASYNC controller supports two asynchronous ports. At least two ASYNC controllers are required for the OSP and for redundancy. Up to two ASYNC extension boards may be added to each ASYNC controller, if needed.
21. One SYNC patch panel is required for each BYTE SYNC controller. SYNC controllers are ordered in pairs for redundancy.
22. No SYNC patch panels are ordered for BIT SYNC controllers.
23. Communications patch panel/line monitor and ARCLI components are never ordered.
24. One FOX controller is required per node. A single FOX cable connects two nodes.

2. Software

Unique configuration heuristics described below apply to software line items.

1. All FDC software is purchased on a "per site" basis (i.e., pay for the first copy only at any site) and

ordered on a "per processor" basis. This requirement includes Batch, FDC System Utilities, FDC File Security System, FDC TPS SAS, System Card Reader Support and GFE Terminal Support packages.

2. TANDEM software is purchased and ordered on a "per processor" basis. This requirement includes GUARDIAN OS, ENCOMPASS, EXPAND and COBOL packages. TANDEM EXCHANGE RJE HASP software can not be ordered.
3. All 6100 software is ordered on a "per processor" basis. 6100 software versions must be indicated when ordering since versions differ for each site.
4. DDN Service Interface software is ordered on a "per site" basis. DDN Interface Protocol software is ordered on a "per processor" basis.
5. NETEX software packages (feature numbers 550801 through 551302) do not have any warranty period. No maintenance uplift factor should be applied to these software packages. NETEX software ordered will correspond to the NSC HYPERchannel adapters ordered. Pricing for Burroughs NETEX software is set at the maximum discount level. Pricing for TANDEM NETEX software is set at the third level. Pricing for all other NETEX software products are set at the first level.
6. Software maintenance is computed on a "per site" basis.
7. Block Structured Language (PASCAL) and FORTRAN may only be ordered for FMSO Sites 02 and 03.
8. Software components which are part of a bundled package may not be ordered separately.
9. FMSO Configuration Management and Query software may not be ordered.
10. T-TEXT software must consciously be ordered.

3. Manuals and Documentation

Four sets of manuals are available on the SPLICE contract. A predetermined number of manuals has been identified for each site. This predetermined figure is an

element of the input configuration file. Nevertheless, the actual number of manuals desired for a site must be specified during configuration processing. This is necessary since sites may not require the predetermined quantity on the first delivery.

4. Training

Training was originally planned to be ordered on a group basis. Several individual courses may be ordered either in addition to or in lieu of the group package. Such an option is supported for the following courses:

1. Hardware Overview
2. Systems Resource Management
3. Systems Tuning and XRAY
4. Data Communications
5. TANDEM Applications Language (TAL)

The addition of courses in the future will require the modification of source code and the input cost data file. This action will only apply to courses ordered on a unit basis.

5. Maintenance

Maintenance is configured on a component and monthly unit basis with few exceptions. If the normal maintenance option is selected, preventive maintenance and on-call maintenance options have zero values for both quantity and cost. If the normal maintenance is not selected, preventive

and on-call maintenance options are assigned values according to the SPLICE contract. Emergency Per-Call maintenance is specified on an hourly basis. Months of component maintenance varies based upon the warranty period specified in the SPLICE contract.

6. Other

Site Preparation (initial site preparation and installation survey) charges must be specified during the configuration process if desired.

7. Discount and Escalation Rates

Discount and escalation rates specified in the SPLICE contract vary at predetermined levels. These rates vary based upon either elapsed time relative to the contract award date or the quantity of line items ordered. The discount and escalation rates applied to line items during the configuration process must be explicitly specified. The rates entered are added to a value of one to generate the appropriate multiplication factor. Discount rate entries must be entered as negative amounts. The multiplication factor is then applied to a basic rate obtained from an input cost data file.

The heuristics described above apply to contract line items of a fifteen year life cycle ADP contract. As ADP technology is ever and rapidly changing, new requirements and pricing options are negotiated between the

government and the vendor (FDC). Accordingly, modifications to these heuristics will be necessary on a continual basis.

III. METHODOLOGY USED TO DEVELOP THE SYSTEM

The idea to pursue the development of a micro-computer knowledge-based configuration system was fostered by the need to satisfy a group project for a course of instruction in decision support systems (DSS). A member of the group was the former FMSO SPLICE project manager. Familiar with the specifics of the SPLICE project and sensitive to the problems experienced by the NAVSUP SPLICE project management staff, he proposed the development effort. Development of the proposed system would satisfy two purposes:

1. the need to complete a group project for the DSS course
2. provide an automated micro-computer knowledge-based configuration system that would help alleviate some of the NAVSUP SPLICE project staff's work load. Additionally, the proposed system would yield a more accurate, consistent and reliable configuration process.

The initial proposal was to develop a knowledge-based configuration system. No follow on development was planned as part of the initial development. TURBO Pascal was selected as the programming language of choice for the following reasons:

1. all group members were familiar with the language as a result of exposure from a previous programming course
2. a structured programming language was desired for the development effort

3. a language which supported screen-oriented functions and color was desired
4. a language which provided quick response and ease of editing and compilation to reduce development effort and minimize frustration

Other programming languages could have satisfied item 2 through 4 requirements as well, but TURBO Pascal was chosen because of the overriding requirement of item 1. This requirement was felt to be of paramount importance due to the short development time frame involved for the course. Group members felt that familiarity with TURBO Pascal would allow the development effort to be modular and completed more rapidly. The system was completed and was forwarded to NAVSUP for evaluation and comment.

A follow on course of instruction dealing with software engineering methodologies was taken. A course requirement called for the development of a project using a structured software engineering approach to software development. Feedback from the NAVSUP SPLICE project staff was favorable. Comments received indicated a strong potential for the system to significantly improve the currently manual configuration process. Follow on group development of the project was initiated. The group discussed the merits of such a system and decided to pursue development employing the software engineering methodology taught in the course. Discussion for the remainder of this chapter will focus on the entire development effort from commencement of

development to completion of the integrated configuration management system.

A. PROTOTYPE

During the initial discussions and planning of the proposed configuration system, the major concern of group members was whether the vast number of heuristics involved in the configuration process could successfully be automated during the time frame of the course. In order to meet the completion deadline, the programming effort had to be divided between group members. The strategy employed was to break the system down into five basic functional areas. Each functional area would deal with each set of heuristics described in the previous chapter with only minor exceptions. The general heuristics had to be addressed for multiple areas and a few of the smaller areas were consolidated for development efficiency.

The group strategy was to start with the first group of heuristics (hardware) and proceed in an incremental fashion. Development effort would continue until either the prototype system was finished or until the project was due. Since there were so many heuristics involved and no formal structured design or engineering methodology was conducted, there was little certainty of how much of the system would be developed.

Development commenced with the general and hardware heuristics. Initially, development was extremely slow and difficult. General and hardware heuristics encompass the majority of the heuristics associated with the configuration process and are very complex. The incorporation of these areas into the system consumed the largest amount of time during the prototype development effort. Development continued sequentially by area until all areas had been addressed. As each area was implemented, development became easier as members gained confidence and heuristics became less complicated.

As mentioned in the introduction, the initial goal in the development effort was to make the system interactive and as user friendly as possible. The screen oriented features and functions of TURBO Pascal proved to be very beneficial in this endeavor. The use of colors for screen displays helped to differentiate input fields and prompts. The ability to move the cursor anywhere on the screen and control data entry, validation and error messages formats also aided in this effort.

Upon completion of the course, the prototype configuration system was forwarded to the NAVSUP SPLICE project staff for comments and recommendations. Project staff personnel expressed considerable interest in the prototype configuration system. While the configuration system was crude, project staff personnel were enthusiastic

about the potential benefits of the system. Discussions concerning their desire to incorporate other project management functions into the system were addressed.

B. SOFTWARE ENGINEERING METHODOLOGY

The software design course requirement to develop a software system using a structured methodology coincided closely with the receipt of the NAVSUP list of comments, recommendations and additional features. Further development of the system was accomplished using a programming team concept in conjunction with the software engineering methodology.

The software engineering methodology used in the development effort is a three phased structured approach encouraged by Pressman:

1. Planning - the definition, analysis, specification, estimation and review of a process. Planning provides a preliminary indication of project viability in relationship to cost and schedule constraints
2. Design - a process of applying various techniques and principles for the purpose of defining a device, a process, or a system in sufficient detail to permit its physical realization
3. Maintenance - the diagnosis and correction of errors (corrective); the modification of software to properly interface with a changing environment (adaptive); or the incorporation of recommendations for newer capabilities, modifications of existing functions, or general enhancements following the successful development of software (perfective)

Each phase of the structured methodology is designed to minimize the difficulties associated with the software development effort. [Ref. 2]

1. Planning

The first step of the software engineering methodology is the planning process. During this phase of software development, the group commenced the detailed planning of the functions that were to be incorporated into the system. Initial discussions centered around the level of complexity to be attempted for the course project. During these discussions, comments, recommendations and additional features provided by the SPLICE project staff were reviewed and scoped for level of complexity.

Initial planning efforts generated a proposal to develop an integrated interactive and user-friendly system that would be composed of three major functional modules:

1. Configuration module
2. Financial analysis module
3. Configuration Management System module that would support report generation

Detailed functions for each module were further specified. Individual member previous experience and strengths were evaluated. The group was organized into a programming team concept. Each member was assigned tasks which best corresponded to his level of experience and knowledge with respect to development tasks.

Once the system functional modules were identified, the next step involved the selection of software to implement the development effort. Based upon the effort that had been expended and the enthusiasm exhibited with the prototype development, a decision was made to continue development of the configuration module using TURBO Pascal. SCREEN SCULPTOR³ was selected for the purpose of developing customized screens for the configuration module. It also employed a data entry and validation feature that could be incorporated into the configuration module with little effort. LOTUS 1-2-3 was selected as the software package for development of the financial analysis package. This selection was based upon the fact that the package was owned by a member of the group who was familiar and experienced in its use. dBASE III was selected for development of the Configuration Management module. Reasons surrounding this choice were:

1. the package was owned and readily available
2. it could be used as a shell to call and run other software packages from as well as perform the functions of configuration management using data base technology

³SCREEN SCULPTOR is a software product available from The Software Bottling Company of New York. 6600 Long Island Expressway, Maspeth. NY 11378 (718) 458-3700. SCREEN SCULPTOR is a programming productivity tool that enables programmers to design and create input screens in minutes in either BASIC, IBM Pascal or TURBO Pascal.

3. FLASH CODE,⁴ a commercial screen generation software package was available and could support the generation of customized screens and perform data entry validation for both dBASE II and dBASE III. The use of such a package would help minimize development effort and ensure correct data entry
4. dBASE III could support ten open files concurrently
5. no other data base management software package was available that either provided the capability to customize screens to the degree desired and support an interface to FLASH CODE

WORDSTAR was selected as the word processing software package that would be used to enable the user to view the User's Manual on-line. All packages with the exception of the two screen generation development packages were currently being used by SPLICE project staff personnel and required little investment in time to learn new packages or the outlay of funds.

Selection of the software packages posed some problems which had to be overcome prior to further development. LOTUS 1-2-3 and dBASE III both required special file formats and interfaces between input and output of each functional module. Special conversion procedures had to be developed to overcome these interface difficulties.

⁴FLASH CODE is a software product available from The Software Bottling Company of New York, 6600 Long Island Expressway, Maspeth, NY 11378 (718) 458-3700. FLASH CODE is a programming productivity tool that provides dBASE II or dBASE III programmers the capability to use either screens or pop-up windows/help menus that instantaneously flash up on the screen.

The Pascal configurer module had to be developed to generate an output file that would allow the viewing and processing of both text and numerical fields when imported into LOTUS 1-2-3. The output file from the LOTUS 1-2-3 financial analysis module stripped off all text and header data following financial verification and saved as a ".PRN" data file. A dBASE III work data base had to be created using a structure that was compatible with the ".PRN" data file. This ".PRN" file was later appended to the dBASE III work data base and converted to a dBASE III data entry format.

With the module interfaces resolved, each functional module was further developed and refined to identify all data elements involved with the functional process. Data flow diagrams documenting all required data elements and processes were generated for each functional module. Two data flow diagrams are provided in Appendix B to serve as representative examples of this process. Each data flow, input file and functional process was further specified in detail through the use of various module descriptions. An example of each of these description modules is provided in Appendix B. The formats of each of the descriptions used in the definition process were modifications of formats specified in [Ref. 2] and [Ref. 3]. A Bachman diagram, supplied in Appendix B, was used to document the data base relationships associated with the configuration management

module. The generation of all functional module data interdependency charts signaled the completion of the planning phase.

2. Development

With the definition of all data element relationships, interdependencies and functional interfaces defined, the group commenced the development phase of the methodology. Using the data flow diagrams, data flow and process descriptions generated during the planning phase, each data process or bubble was decomposed into more detailed sub-functional processes.

Sub-functional processes were developed by exploding each bubble from the data flow diagram and decomposing the process to its lowest functional level through several layers of abstraction. The lowest levels of abstraction are procedure oriented and are stated in terms that can be directly implemented. Several guidelines for the process are involved and are outlined in Pressman [Ref. 2]. The overall objective of this decomposition process was to arrive at a description of each functional process to a level that would support modular development. Appendix B contains a few structure charts which are representative examples of the decomposition process.

The idea behind decomposing each process to its lowest functional description is to ensure that the scope of

effect⁵ of a module is maintained within the scope of control⁶ of that module [Ref. 2]. Another concept of the engineering methodology designed to aid in the development and maintenance of software systems is that of information hiding⁷. These concepts were applied to the design phase of development to ensure modularity of the system. The structure of the system was designed in a way that would facilitate future maintenance.

With all processes defined, team members began coding the various modules. Coding was accomplished in a top-down modular fashion to facilitate a phased implementation plan. As each module was completed, it was integrated into the overall system and tested to ensure accurate performance. Coding continued until the project was due for submission. At the end of the course, the configuration and financial analysis modules were complete. The third module, the data base configuration management system, had a basic structure that would support a minimal number of configuration reports. This module would be

⁵Scope of effect of a module is defined as how other modules are affected by decisions which are made within the module. [Ref. 2: p. 170]

⁶Scope of control of a module is the number and degree of control which is exerted on other modules by the controlling module. [Ref. 2: p. 170]

⁷Information hiding is the concept whereby procedures and data information within a module are invisible to other modules. This concept helps achieve modularity during development. [Ref. 2: pp. 156-157]

finished as a follow on project under the maintenance phase. The system was forwarded to the SPLICE project staff for evaluation.

3. Maintenance

The structured design and development methodology employed in the development of the micro-computer knowledge-based integrated configuration management system proved to be very beneficial. Completion of the data base configuration management module was straight forward due to this design methodology.

The data base configuration management system was completed as a follow on project for a course of instruction in data base design. Since a foundation already existed as a result of the initial system development, continued development fell into the category of maintenance. The development of the configuration management module used three methods of maintenance. Each maintenance category is defined briefly in the methodology introductory discussion near the beginning of this chapter.

Continued development of the configuration management module was undertaken. Feedback from the SPLICE project staff highlighted errors which required correction - corrective maintenance. Also, due to contract negotiations and modifications, certain heuristics required modification - adaptive maintenance. Additionally, the data

base design course highlighted more efficient methods of accomplishing functional processes in lieu of methods used during the development phase of the system - adaptive and perfective maintenance.

The maintenance effort and system enhancements proposed by the NAVSUP SPLICE project staff were reviewed and evaluated for level of implementation difficulty. Each change was classified according to the type of maintenance involved. A development schedule was established and development effort continued.

The first maintenance actions addressed were corrective maintenance issues. Each potential error was evaluated in terms of its impact on the basic system structure. Errors were also evaluated in terms of whether the condition fell within the initial capabilities designed for the system. Some of the potential errors were found to be outside the scope of the initial design and were not attempted. SPLICE project staff personnel were informed of these conditions and were instructed on how to deal with the conditions.

Changes to the initial environment were addressed next. Contract negotiations are continuing and result in contract modification requirements. These modifications were evaluated to identify the degree of modification required to the basic system structure. While some modification was required, the majority of the changes

involved the configuration module. The decomposition of the logical functions to their lowest levels coupled with the high degree of cohesion⁸ and low degree of coupling⁹ of both modules and data made maintenance almost effortless.

The last maintenance area involved refining the methods by which tasks were performed. Knowledge gained from the data base design course identified more efficient means of accessing certain files. Also, certain initial relationships did not follow the relational normal forms associated with relational data base design [Ref. 4] and [Ref. 5]. Thus, certain files had to be restructured. Other changes involved eliminating unnecessary statements and optimizing certain functions, loops and file accesses. Modification of certain file accesses resulted in the reduction of response times in some cases by eighty to ninety percent.

Completion of the data base configuration management module marked the final development of the micro-computer knowledge-based interactive configuration management system for the SPLICE project. NAVSUP SPLICE project staff personnel have the system and are currently using the system

⁸Cohesion is a measure of the relative functional strength possessed by a module (i.e. a cohesive module should only perform one thing or function) [Ref. 2: p. 158]

⁹Coupling is a measure of the relative interdependencies between modules (i.e., the degree to which other modules are dependent upon interfaces and data) [Ref. 2: p. 161]

for initial configurations. Once current sites under configuration are loaded to system data bases, sites previously configured will be loaded. The SPLICE project manager now has the capability to configure sites, perform financial and "what-if" analysis and generate a wide variety of reports to aid in the management of the project. The system report generation facility also enables the project manager to track components by serial number and location. The development of the micro-computer knowledge-based interactive configuration management system has provided the SPLICE project manager with the capability not only to evaluate overall project performance, but also to evaluate the contract vendor's performance with regard to contract requirements.

C. SUMMARY

The development of the micro-computer knowledge-based interactive configuration management system involved several different development methodologies. The success of its development could not have been realized without the inclusion of all methodologies.

Prototyping, while not a solution by itself, identified several problems with the original system design and data entry method. It also highlighted several areas which required modification to achieve the goal of developing a user-friendly system.

The execution of the software engineering methodology described by Pressman [Ref. 2] helped to identify all of the functional tasks for logical incorporation into the system. The use of the various module descriptions identified all of the essential data elements, flows and processes. The use of these descriptions further helped to minimize development time and prevent needless rework. Incremental implementation of completed modules kept the development effort on schedule. The use of commercially proven and tested "off-the-shelf" packages further helped to minimize the development effort.

The SPLICE micro-computer knowledge-based interactive configuration management system is an active system. As with any software system, maintenance must be performed to maintain the system current with its operational environment. The SPLICE configuration management system is no different. Due to a changing environment and requests for further enhancements to the system, a backlog of changes currently exists.

Due to the methodologies used in the design and development of the SPLICE configuration management system, the backlog and future changes should be able to be incorporated into the system with minimal confusion or effort.

IV. SYSTEM EXECUTION DIALOGUE

As discussed in previous chapters, the micro-computer knowledge-based configuration management system is an interactive and user-friendly system. Additionally, the system is an integrated system composed of three functionally separate modules:

1. configuration module - developed using TURBO Pascal
2. financial and "what-if" analysis module - developed using LOTUS 1-2-3
3. configuration management and report generation module - developed using dBASE III

Integration of the system was possible through dBASE III's ability to run other programs during system execution. This feature allowed dBASE III to be used as the shell or driver for the system.

Following discussions describe a typical system execution dialogue. All screen formats mentioned or referenced may be found in Attachment 2 of Appendix A. The system has no on-line help facility other than the on-line User's Manual. Review of the User's Manual may only be accomplished from the system's opening menu (Screen 1). Detailed information regarding system execution is addressed in Appendix A.

A. SYSTEM INITIATION

With initial installation complete and the target system's power on, type the command SPLICE at the DOS command prompt to initiate system execution. The first screen viewed is the Function Selection Menu - Screen 1. From this menu, the user may select any one of six possible options.

B. CONFIGURE A SITE

The first function normally performed would be to configure a site for SPLICE installation. This action is accomplished by selecting menu option 1 from the Function Selection Menu. Selection of this option invokes the Pascal Configuration Module. The user, having accumulated the applicable data for the site to be configured and recorded the information on a copy of Attachment 1 of Appendix A, would commence the configuration process.

The user would first see a module logo and version screen (Screen 2) followed by five data entry screens (Screens 3 through 8) and a final output screen (Screen 9) identifying the output file name to be imported into the financial analysis module. The data field sequence of Attachment 1 to Appendix A is in the sequence of data entries expected for screens 3 through 8.

Screen 3 is a list of designated SPLICE sites. Screens 4 through 8 are the applicable data entry screens. Data

entry is segmented into component and data types (ex: discount and escalation rates, hardware, software, etc.). The output data file name is presented as part of the final display to the configuration module (Screen 8). The output file is formatted for data entry into the financial analysis module. Following completion of the configuration process, the user is returned to the Function Selection Menu.

C. PERFORM FINANCIAL ANALYSIS ON SITE DATA

Financial analysis and delivery order preparation is the next function to be performed. Selection of menu option 2 from the Function Selection Menu invokes the execution of the financial analysis module using the LOTUS 1-2-3 system. The output file previously generated from the configuration module may then be viewed.

Several LOTUS macros, described in detail in Appendix A, enable the configuration module calculations and computations to be verified. "What-if" analysis may also be performed to evaluate the impacts of system costs relative to options selected and/or modify a system configuration to coincide with the current funding environment. Screen 13 is a partial example of how the data is presented in the financial analysis module. Upon completion of the configuration analysis, the data file is formatted for input into dBASE III data base files. Following financial

analysis termination, the user is returned to the Function Selection Menu (Screen 1).

D. INTERACT WITH THE CONFIGURATION MANAGEMENT AND REPORT GENERATION SUB-SYSTEM

Execution and interaction with the configuration management and report generation sub-system is invoked by selecting menu option 3 from the Function Selection Menu (Screen 1). The Process Selection Menu (Screen 14) is displayed and reveals nine additional options from which to choose.

1. Load New Delivery Order Data

The most common option to select will be menu option 1 - load the formatted file from the financial analysis module to the various data bases. The process is menu driven requiring answers to a few questions presented on screens 15 and 16. The data loading process adds new records to three data bases. If the input file is very large, the loading process may be lengthy.

Completion of loading data to the three data bases signals the interim completion of the configuration process for a site. No further data for the site may be loaded to the data bases until the equipment is received at the site. From this point, the user may return to the Process Selection Menu and obtain any of several reports extracted

in a variety of formats or return to the Function Selection Menu and choose another processing option.

2. Load Serial Number and Manual Data

Following the receipt of ordered components at the applicable site, the user may load the serial numbers of the hardware components and the names of the accompanying hardware and software manuals received. This function is a two step process.

Serial numbers may be loaded to the serial number data base by selecting menu option 6 from the Process Selection Menu, whereby the Serial Number Maintenance Menu (Screen 32) is displayed. Selection of menu option 1 results in the presentation of the Serial Number Update Format screen (Screen 33). To enter the applicable serial numbers, the user must provide the system with three data elements to load the serial number data:

1. site number
2. effective date of the applicable delivery order
3. feature number of the component

Once all three data elements have been entered, the serial number may then be entered. This process must be iterated for each serial number to be loaded to the data base. Since neither serial number nor manual information is available during the initial data load process, it is necessary to specify all three serial number data elements to ensure data

and file integrity. Following entry of the last serial number, the user terminates the update process by selecting the exit (X) option. This returns the user to the Serial Number Update Format screen (Screen 33). The user may either review the serial numbers just entered or return to the Process Selection Menu to initiate the loading of the applicable manual data.

Following entry of the serial number data, the applicable manual description data may be loaded to the Manual data base. This is accomplished by selecting menu option 5 from the Process Selection Menu, whereby the Manual Maintenance Menu (Screen 27) is displayed. To add manual descriptions to the manual data base, select menu option 1. The Manual Addition Format screen (Screen 28) is displayed. To enter the manual descriptions, first enter the applicable site number followed by the associated feature number for the manual description to be loaded.

Following entry of the last manual description, terminate the addition process by selecting the exit (X) option. This returns the user to the Manual Maintenance Format screen (Screen 27). The user may either review the manual descriptions just entered or return to the Process Selection Menu to initiate another process selection.

3. Generate a Maintenance Delivery Order

At the commencement of each fiscal year, the NAVSUP SPLICE project staff must initiate a delivery order to cover the maintenance and rental services for the current fiscal year for each configured SPLICE site. To accomplish this task, select menu option 8 from the Process Selection Menu (Screen 14). The Maintenance Delivery Order Generation Program screen (Screen 66) is presented and requires five inputs. First, the applicable site number for which the maintenance delivery is to be generated is entered. Then four discount or escalation rates are entered. These rates are based upon pre-determined terms negotiated in the SPLICE contract. These rates are based upon total number of components ordered and the elapsed time relative to the contract award.

A new formatted file (NEWDO.PRN) is generated to be imported into the financial module where computations and calculations are verified in the same manner discussed in section C above. Once the data has been verified financially correct in the financial module, the maintenance delivery order is ready to be printed. Program execution then automatically returns the user back to the Process Selection Menu where another process selection may be made.

4. Generate a Report

A variety of eight different reports are available from the report generation sub-system. Reports are available for:

1. the overall project
2. a particular site
3. a delivery order issued on a particular date

Within these categories, reports may further be broken down by:

- a. equipment type
- b. serial number

Delivery order equipment type reports may be obtained either with or without unit price data in the report.

The generation of any one of the eight available reports is obtained by initially selecting menu option 7 from the Process Selection Menu, whereby the Report by Type Menu (Screen 36) is displayed. Depending on the type of report desired, further menu options are selected. Screens 36 through 65 are examples of the various menus and report formats that are obtainable from the report generation system but are not discussed in detail.

E. REVIEW THE ON-LINE USER'S MANUAL

The on-line User's Manual may be viewed any time the user is viewing the Function Selection Menu (Screen 1). As stated before, no on-line help facility is available during

functional module execution. The on-line User's Manual uses WORDSTAR as the word processing package to display system execution instructions to the user. As such, the ability to jump to a specific page or process description does not exist. Following termination, the user is returned to the Function Selection Menu (Screen 1).

F. TERMINATE SYSTEM EXECUTION

When all system functions have been performed and the user desires to terminate system execution, two options are available. Menu options 5 and 6 on the Function Selection Menu (Screen 1) allow the user to either terminate system execution and return to the dBASE III environment (dot prompt) for further interactive queries or terminate system execution and return to the DOS operating environment. The most common selection will likely be to terminate system execution and return to the DOS operating environment.

V. COST BENEFIT AND EFFECTIVENESS

Prior to the development of the micro-computer knowledge-based integrated configuration management system for the NAVSUP SPLICE project staff, the first eight of a possible sixty-two initial site configurations were processed in a semi-automated fashion. While LOTUS 1-2-3 was used as the medium to produce the final form delivery order, a considerable amount of the heuristic processing still was manual. The developed system eliminates all such manual processing, except for gathering the initial sizing study input data.

Within the NAVSUP SPLICE project staff, one mid-grade GS-12 government employee is currently responsible for all SPLICE site configuration processing, project configuration management and vendor contract performance monitoring. Average annual salary for this grade level for a step five position is approximately thirty-six thousand dollars.

In the current phase of the project life cycle, sites are being configured for their initial equipment and associated software components. Existing sites with initial configurations require maintenance delivery orders generated to support continuing maintenance services on an annual basis. As mentioned in the introduction, errors discovered in delivery orders submitted to the vendor for processing

are corrected, with an additional charge¹⁰ levied upon the government for the additional service. Due to the minimum number of sites that have been configured and are in operational status, there currently is little configuration management being performed.

To evaluate the benefit and effectiveness of the developed system, certain (worst case) assumptions are made:

1. based upon previous experience, each delivery order supplied to the vendor will contain errors
2. the government will incur a five thousand dollar additional charge for vendor corrections to **initial configuration** delivery orders containing errors
3. the government will incur a one thousand dollar additional charge for vendor corrections to **maintenance** delivery orders containing errors (no experience exists to evaluate the accuracy of this assumption and is therefore an anticipated worst case assumption)

Since only a few of the designated sites are currently operational, the one GS-12 employee has managed to keep pace with the work load. Without the development of the micro-computer knowledge-based integrated configuration management system, this effort would not be possible and

¹⁰Charges of up to five thousand dollars per delivery order to correct existing errors have been experienced.

would most likely require the hiring of another lower grade employee on a full time basis¹¹ in the future.

During the next two calendar years, the remaining initial site configurations are going to be processed.¹² Figures based on the worst case assumptions stated above, suggest that the developed system has the potential to yield savings of close to two-hundred and fifty thousand dollars for the initial configuration process alone. Since each site must have a maintenance delivery order generated each fiscal year to account for increases or decreases in maintenance rates for services, the potential exists to realize additional savings of approximately sixty thousand dollars for each remaining year of the project life cycle.

The SPLICE contract contains predetermined discount and escalation rates which were negotiated and written into the contract. Certain discounts depend upon the quantity of components previously ordered and are graduated according to predetermined procurement levels. The ability of the GS-12 employee to currently identify these discount levels is

¹¹ Once all SPLICE sites have been configured for initial equipment and component installation, configuration management within the project will come to the forefront. Due to the large number and variety of components that may exist for any site which can have an impact on the discounts that are applicable to component, this phase of contract monitoring and execution becomes critical in terms of cost effectiveness.

¹² Approximately twenty sites are scheduled for configuration during CY 1986 and approximately thirty sites are scheduled for configuration during CY 1987

accomplished solely through a manual process. Each delivery order previously issued has to be manually totaled to arrive at each component's project procurement total. Through the developed system's report generation facility, potential discounts can be identified in a matter of seconds. The potential savings that may be realized in this manner are difficult to quantify. I feel that it is safe to say that over the life cycle of the project, substantial savings as a result of this new capability can result.

The developed system provides the NAVSUP SPLICE project staff with the ability to monitor the vendor's performance relative to contract specifications and perform configuration management for the overall project. While the contract provided a configuration management package line item for these services, development of the system precludes the need to procure the option priced at roughly one-hundred thousand dollars.

The developed system provides the project staff with extensive capabilities needed to properly execute their functions as overseers of the contract and does so in an automated and efficient manner. These capabilities are believed to be developed to a level that will allow the existing project staff employee to perform these functions in roughly half the time experienced prior to system implementation. This increased efficiency should realize a

minimum savings of approximately eighteen thousand dollars each year for the project staff budget.

As seen from the above analysis, the development and implementation of the micro-computer knowledge-based integrated configuration management system for use by the NAVSUP SPLICE project staff provides a more efficient method with increased capability to effectively execute project manager responsibilities and monitor vendor performance. Potential savings realized through the use of this system will be at least eighteen thousand dollars annually for the next few years with the potential to save two-hundred and fifty thousand in the initial configuration process and sixty thousand dollars in annual maintenance modifications.

LIST OF REFERENCES

1. Naval Supply Systems Command Letter UNCLASSIFIED 0472/DAF 5232 Serial 1485U to Naval Data Automation Command (NAVDAC 12). Subject: Request for Life Cycle Management (LCM) Milestone III Approval of the Stock Point Logistics Integrated Communications Environment (SPLICE) Project (U), Enclosure (2). pp. 2-1 to 2-5. 1 March 1985.
2. Pressman, Roger S.. Software Engineering. McGraw-Hill, Inc. 1982.
3. Yourdan, E., and Constantine, L., Structured Design, Prentice-Hall, 1979.
4. Kroenke, David M.. DATABASE PROCESSING: Fundamentals, Design Implementation, 2d ed., pp. 286-330, Science Research Associates, Inc., 1983.
5. Kent, William. "A Simple Guide to Five Normal Forms in Relational Database Theory," Communications of the Association for Computing Machinery, Vol. 26, No. 2, February 1983, pp. 120-125

APPENDIX A

THE NAVAL SUPPLY SYSTEMS COMMAND
STOCK POINT LOGISTICS INTEGRATED COMMUNICATIONS ENVIRONMENT
(SPLICE)
SYSTEM CONFIGURER AND CONFIGURATION MANAGEMENT SYSTEM
USER'S MANUAL

Document No. BBC - 01

1 January 1986

Record of Changes

Original

1 January 1986

List of Effective Pages

Page 1 through 44	Original
Page A1-45 through A1-49	Original
Page A2-50 through A2-83	Original
Page A3-84	Original

Acknowledgements

1-2-3 is a registered trademark of Lotus Development Corporation.

dBASE II and dBASE III are registered trademarks of Ashton-Tate.

Flash Code and Screen Sculptor are registered trademarks of The Software Bottling Company of New York.

Turbo Pascal is a registered trademark of Borland International

Wordstar is a registered trademark of MicroPro International Corporation.

Zerodisk is a registered trademark of Quaid Software Limited.

Recognition

The development of the integrated SPLICE Configuration Management System involved several people. The effort devoted to the finished product was spread over a nine month period. The system was also used to satisfy project assignments in several core courses leading to the receipt of the Master of Science degree. Recognition is acknowledged for the persons listed below for their participation in the completion of the SPLICE Configuration Management System.

Major John P. Barrett, U. S. Marine Corps - test plan generation and tester.

Lieutenant Commander Robert L. Beard III, Supply Corps, U. S. Navy - Pascal and dBASE III programmer, screen generation integrator, User's Manual co-author.

Lieutenant Commander Winston H. Buckley, Supply Corps, U. S. Navy - programming team librarian, documentation generation.

Lieutenant Commander Edward J. Case, Supply Corps, U. S. Navy - system concept, designer, and integrator, Pascal, LOTUS 1-2-3 and dBASE III programmer, User's Manual co-author.

Lieutenant Commander Gary R. Harmeyer, Nurse Corps, U. S. Navy - dBASE III programming assistant, document generation.

Major David L. Horton, U. S. Marine Corps - initial data base designer, dBASE III programmer.

Table of Contents

Record of Changes -----	2
List of Effective Pages -----	3
Acknowledgements -----	4
Recognition -----	5
Table of Contents -----	6
Introduction -----	9
Background -----	9
Why The System Configurer and Configuration Management System -----	9
Input Data -----	11
SPLICE System Configurer and Configuration Management System Files -----	11
System Preparations -----	14
System Execution -----	14
FUNCTION 1: Execute the Pascal Configurer -----	15
FUNCTION 2: Perform LOTUS 1-2-3 Financial or "What-If" Analysis -----	18
FUNCTION 3: Execute the dBASE III Configuration Management System -----	23
1. Load a new Deliver Order into the Configuration Management System -----	24
2. Perform maintenance on the Equipment File -----	25
a. Modify an Equipment File record -----	25
b. Review an Equipment File record -----	26
3. Perform maintenance on the Equipment Description File -----	27

a. Modify an Equipment Description File record -----	27
b. Review an Equipment Description File record -----	28
4. Perform maintenance on the Site Name File -----	28
a. Modify a Site Name File record -----	29
b. Review a Site Name File record -----	29
5. Perform maintenance on the Manual File -----	30
a. Add a new manual description record -----	30
b. Update a manual description record -----	31
c. Delete a manual description record -----	32
d. Review a manual description record -----	32
6. Perform maintenance on the Serial Number File -----	33
a. Modify a Serial Number File record -----	33
b. Review a Serial Number File record -----	34
7. Generate reports for the Project, a specific site or a particular date -----	35
a. Overall Project reports -----	35
(1) Reports by Equipment type -----	36
(2) Reports by Serial Number -----	36
b. Reports for a Particular SPLICE Site -----	37
(1) Reports by Equipment type -----	37
(2) Reports for Manuals -----	38
(3) Reports by Serial Number -----	38
c. Report for a Delivery Order issued on a particular date -----	39
(1) Report by Equipment type with unit prices -----	39

APPENDIX A: USER'S MANUAL

Page 8

(2) Report by Equipment type without unit prices --	40
(3) Report by Serial Number -----	41
8. Generate a Maintenance Delivery Order for a specific SPLICE site -----	41
9. Generate Mailing Labels for all SPLICE sites -----	42
FUNCTION 4: View the on-line User's Manual -----	43
System Output -----	43
Exception Reports -----	43
Limitations -----	43
Command Sequence -----	44
Who To Call -----	44

Attachments:

1. Selection Criteria for SPLICE Configurations ----	A1-45
2. Screens Formats -----	A2-50
3. Installation Procedures -----	A3-84

1.0 Introduction.

This manual is designed to provide information and guidance to the SPLICE integrated system user. The integrated system components include: 1 - the SPLICE System Configurer, 2 - the LOTUS 1-2-3 financial and "what-if" analysis system, 3 - the dBASE III Configuration Management System, and 4 - the Wordstar on-line User's Manual.

1.1 Background.

The Naval Supply Systems Command (NAVSUP) conceived and developed the Stock Point Logistics Integrated Communications Environment (SPLICE) project. The SPLICE project purpose is to:

- a. Provide state-of-the-art local and long haul telecommunications capabilities to 62 NAVSUP Stock Points.
- b. Provide interactive and distributed ADP processing capabilities to SPLICE sites.
- c. Provide capacity relief to aging Burroughs hosts at the Stock Points.
- d. Standardize and upgrade, via mass replacement, the myriad of minicomputers existing at Stock Points.

NAVSUP initiated a competitive solicitation for "fault-tolerant" hardware and software to achieve these goals. The solicitation was completed in November 1983. The winning vendor, Federal Data Corporation (FDC), proposed TANDEM hardware and software to meet most of the solicitation processing and local communications requirements. FDC proposed Network System Corporation hardware and software to meet the local inter-host communications requirements.

1.2 Why The System Configurer and Configuration Management System.

Shortly after the SPLICE contract award, hardware and software components had to be ordered. NAVSUP faced a dilemma. Only a few SPLICE personnel had worked closely with the SPLICE acquisition benchmark and negotiations. These few people were the only personnel that had sufficient

knowledge of the systems to configure and generate delivery orders. These personnel developed initial orders by hand to meet the immediate need. Many minor errors were encountered with these initial orders. FDC corrected and returned the orders to the government and received additional compensation for their efforts.

This manual configuration process was later automated using a software product called SUPERCALC2. It has subsequently transitioned to LOTUS 1-2-3. The basic method of developing these orders remained virtually manual. These few SPLICE personnel, with FDC assistance, developed a series of "rules of thumb" used to configure individual site systems. Many of the original SPLICE group have moved on, taking their knowledge of the systems with them.

This SPLICE Configurer and Configuration Management System software is a knowledge based system designed to codify these "rules of thumb". This integrated system will enable NAVSUP to develop and maintain SPLICE configurations and delivery orders and perform configuration management on the project. Three software products were created in this phase of development to:

- a. Configure initial SPLICE site systems by answering a series of questions. SPLICE.COM (written in TURBO Pascal) produces structured delivery orders that must be imported into LOTUS 1-2-3. LOTUS 1-2-3 performs financial review and analysis before loading the dBASE III Configuration Management data bases.
- b. Restructure the SPLICE.COM output file into LOTUS 1-2-3 format. A series of macros assist in the regeneration of the delivery order into LOTUS standard formula format. Following the conversion, three options exist: 1 - print the delivery orders, 2 - prepare archival files, or 3 - prepare the output file needed for the dBASE III Configuration Management system.
- c. Restructure the LOTUS 1-2-3 output file into dBASE III format. dBASE III command language modules import and convert the LOTUS output file into dBASE III format. They also either generate or update the three dBASE III Configuration Management data bases. This allows the user to generate selected configuration management reports from the three data bases. MAINTDO.PRG, a dBASE III module, generates maintenance delivery orders from the configuration management data bases. These maintenance delivery orders

must be imported into LOTUS 1-2-3 for final financial review and analysis.

2.0 Input Data.

The following paragraphs describe the integrated system data input requirements. The following discussion describes the files required to execute the system and the associated screen formats.

2.1 SPLICE System Configurer and Configuration Management System Files.

The SPLICE System Configurer and Configuration Management System can only be run on a hard disk system, with the following minimum files (refer to Attachment 3 for system installation procedures):

GROUP 1 FILE-IDs (SPLICE Configurer)

- | | |
|---------------|---------------|
| a. COSTS.IN | b. CONFIG.SIT |
| c. SPLICE.COM | d. SPLICE.SCR |

GROUP 2 FILE-IDs (LOTUS 1-2-3 Financial Analysis)

- | | |
|-----------------|--|
| e. 123.EXE | (Associated files for LOTUS version 1A not shown but are also required.) |
| f. SKELETON.WKS | g. MAINTORD.WKS |

GROUP 3 FILE-IDs (dBASE III Configuration Management System)

- | | | |
|----------------|---|-----------------|
| h. DBASE.COM | (Associated files for dBASE III version 1.1 not shown but are also required.) | |
| i. CONFIG.DBF | j. CONFIG.NDX | k. CONFMOD.PRG |
| l. CONFREV.PRG | m. CONFUPD.PRG | n. DATERPTS.PRG |
| o. DELAY.PRG | p. DESCRIP.DBF | q. DESCRIP.DBT |

APPENDIX A: USER'S MANUAL

Page 12

r. DESCRIP.NDX	s. DESCRIPT.SCR	t. DESPMOD.PRG
u. DESPPREV.PRG	v. DESPPUPD.PRG	w. EFEAT.NDX
x. EQPDTNPC.PRG	y. EQPDTPRC.PRG	z. EQPPJRPT.PRG
aa. EQPSTRPT.PRG	bb. EQUIP.DBF	cc. EQUIPCMD.PRG
dd. EQUIPDAT.NDX	ee. EQUIPPRJ.NDX	ff. EQUIPREV.PRG
gg. EQUIPREV.SCR	hh. EQUIPSD.NDX	ii. EQUIPSIT.NDX
jj. EQUIPUPD.PRG	kk. EQUIPUPD.SCR	ll. FLASHUP.COM
mm. MAINMENU.PRG	nn. MAINMENU.SCR	oo. MAINTDO.PRG
pp. MAINTDO.SCR	qq. MANUAL.DBF	rr. MANUALS.SCR
ss. MANULADD.PRG	tt. MANULCMD.PRG	uu. MANULDEL.PRG
vv. MANULREV.PRG	ww. MANULSIT.NDX	xx. MANULUPD.PRG
yy. MKLABELS.PRG	zz. MKLABELS.SCR	aaa. MNLSTRPT.PRG
bbb. NEWDOADD.PRG	ccc. NEWDOCMD.PRG	ddd. NEWDOCVT.PRG
eee. NEWDOCVT.SCR	fff. PROJRPPTS.PRG	ggg. REPORCMD.PRG
hhh. REPORTS.SCR	iii. SELECTOR.PRG	jjj. SELECTOR.SCR
kkk. SERIALNO.DBF	lll. SERIALNO.SCR	mmm. SERNOBLD.PRG
nnn. SERNOCMD.PRG	ooo. SERNODAT.NDX	ppp. SERNOFEA.NDX
qqq. SERNOPRJ.NDX	rrr. SERNOREV.PRG	sss. SERNOSIT.NDX
ttt. SERNOUPD.PRG	uuu. SITENAME.SCR	vvv. SITERPTS.PRG
www. SNODTRPT.PRG	xxx. SNOBJRPT.PRG	yyy. SNOSTRPT.PRG
zzz. SPLICE.BAT	aaaa. SPLICE.WIN	bbbb. TED.DBF
cccc. NEWJOIN.DBF		

Several of the dBASE III command language modules require considerable time to execute. An IBM-PC/XT operating with a clock speed of 6 MHz or greater or IBM-PC/AT provides better performance.

Three additional TURBO Pascal source code files are provided since the Configurer system was developed in Borland International's TURBO Pascal and Software Bottling Company's SCREEN SCULPTOR¹:

GROUP 1 FILE-IDs

a. SPLICE.PAS

b. SPLICE1.PAS

c. SPLICE2.PAS

GROUP 1 files must reside on a subdirectory named \TURBO. GROUP 2 files must reside on a subdirectory named \LOTUS. Group 3 files must reside on a subdirectory named \DBASEIII. The file USERS.MAN must be present on a subdirectory named \WORDSTAR if the User's Manual is viewed on-line (Function Selection Menu option 4). A version of WORDSTAR must also exist on the subdirectory.

Software Bottling Company product FLASH CODE² must be purchased to run the dBASE III Configuration Management System. All command language modules in the dBASE III Configuration Management System use a memory resident program FLASHUP.COM. FLASHUP³ gives dBASE III the extra capabilities of instantly flashing up screens and instantly popping up windows. Load this command module into the computer memory before running dBASE. The SPLICE.BAT

1 SCREEN SCULPTOR is a software product available from The Software Bottling Company of New York, 6600 Long Island Expressway, Maspeth, NY 11378 (718) 458-3700. SCREEN SCULPTOR is a programming productivity tool that enables programmers to design and create input screens in minutes in either BASIC, IBM Pascal or TURBO Pascal.

2 FLASH CODE is a software product available from The Software Bottling Company of New York, 6600 Long Island Expressway, Maspeth, NY 11378 (718) 458-3700. FLASH CODE is a programming productivity tool that provides dBASE II or dBASE III programmers the capability to use either screens or pop-up windows/help menus that instantaneously flash up on the screen.

3 FLASHUP is a memory resident program supplied with FLASH CODE that enables dBASE II or dBASE III programmers to use screens and pop-up windows/help screens which instantly flash up on the screen rather than the dBASE painting method.

command batch file automatically accomplishes this process. FLASHUP is licensed to individuals for use along with either dBASE II or dBASE III programs and may be moved from one computer to another. Any number of people may use FLASHUP, providing there is no possibility of using it concurrently in two or more locations.

Both Software Bottling Company products, **SCREEN SCULPTOR** and **FLASH CODE** must be purchased to perform system maintenance on system screens and windows.

2.2 System Preparations.

Fill out a copy of Attachment 1 before executing the SPLICE Pascal Configurer and Configuration Management System modules. Having this information before beginning a session will greatly facilitate system use.

Turn on the IBM-PC AT target system and the 132 column printer's power. Ensure that the minimum required software listed above is loaded on the active hard disk subdirectories specified. Make subdirectory \DBASEIII the default directory.

2.3 System Execution.

Execute the SPLICE Pascal Configurer and Configuration Management System by entering the command SPLICE at the system prompt (ex: C>SPLICE).

Several copyright notices will appear on the screen after a few seconds delay for system startup. The processes described below are then available: (See Attachment 2 for screen formats).

Screen 1: The Function Selection Menu is the opening screen for the integrated system. Six options exist from which to choose. Option 1 permits the configuration of a SPLICE site. Option 2 uses LOTUS 1-2-3 to perform financial or "what-if" analysis. Option 3 opens the dBASE III SPLICE Configuration Management System. Option 4 reviews the User's Manual on-line. Option 5 returns the system to the dBASE III system prompt. Option 6 returns the system to the DOS prompt. The following discussion is limited to options 1 through 4. Only entries in the range 1 - 6 are valid. The default value is 1.

2.3.1 FUNCTION 1: Execute the Pascal Configurer

Select option 1 (from the Function Selection Menu - Screen 1) to configure a SPLICE site. If the Function Selection Menu is not displayed, select the "Return to " option of the current menu until the Function Selection Menu appears. If a process is active, select the option that terminates the process. Once a menu appears, select the "Return to " option of the current menu until the Function Selection Menu appears. Select option 1 when the Function Selection Menu appears. The first screen of the SPLICE Pascal Configurer (Screen 2) appears.

Screen 2: The opening screen of the Pascal configurer module requires no input.

Screen 3: A list of sites which may be configured appears. Insert an integer value between 01 and 58 to select a currently designated site. Site numbers 59 through 62 are reserved for future designation. Site Number 23 (NAS Oceana) is deactivated and no longer is a designated SPLICE site.

Screen 4: Enter the discount and escalation rates, output file name, number of months of maintenance, and effective delivery order date. Data input ranges apply as described below:

- a. FDC SNA Interface Discount Rate: 0.00 - 9.99
- b. Non-LCN Purchase Discount Rate: 0.00 - 9.99
- c. LCN Purchase Discount Rate: 0.00 - 9.99
- d. SPLICENet Software Maintenance Discount
Rate: 0.00 - 9.99
- e. SPLICENet Software Purchase Discount
Rate: 0.00 - 9.99
- f. Emergency Maintenance Escalation Rate: 0.0 - 9.9
- g. LCN Hardware Maintenance Escalation
Rate: 0.000 - 9.999
- h. LCN Software Maintenance Escalation
Rate: 0.000 - 9.999

- i. Installation Escalation Rate: 0.000 - 9.999
- j. Training Escalation Rate: 0.00 - 9.99
- k. Documentation Escalation Rate: 0.00 - (-9.99)
- l. Maintenance Escalation Rate: 0.000 - 9.999
- m. Output file name: any 8 alphanumeric characters
- n. Hardware Maintenance Months: 0 - 12
- o. Effective Date: 01/01/84 - 12/31/99

On entry of the effective date, confirm the input values by entering a "Y" to the prompt " Do you accept the input values thus far? Yes or No ". The Default value is "N".

Screen 5: Enter the hardware quantities suggested by the Navy Fleet Material Support Office Sizing Study, as transcribed to Attachment 1. The following data input ranges apply:

- a. Processors: 0 - 256
- b. Centronics Printers: 0 - 12
- c. TANDEM CRTs: 0 - 999
- d. 128 MB Disks: 0 - 128, in **EVEN** quantities
- e. 240 MB Disks: 0 - 128, in **EVEN** quantities
- f. 540 MB Disks: 0 - 128, in **EVEN** quantities
- g. Non-6100 ASYNC Controllers: 0 - 64. There should be **at least two in the initial order** for each OSP; subsequent quantities are at the user's discretion.
- h. Non-6100 ASYNC Extension Boards: 0 - 2
- i. Bit SYNC Lines: 0 - 128
- j. Byte SYNC Lines: 0 - 128
- k. Tri-Density Tape Drives: 0 - 128

- l. Reader/Punches: 0 - 12
- m. Card Readers: 0 - 12
- n. 1000 LPM Printers: 0 - 16
- o. 600 LPM Printers: 0 - 16
- p. LCN Coaxial Cables (Trunks): 0 - 2. Input is only allowed for sites designated in file CONFIG.SIT as Stock Points (S).
- q. 6100 Line Interface Units (LIUs): 0 - 256
- r. LCN Interface Adapters (multiple entries): 0 - 256. Input is only allowed for sites designated in file CONFIG.SIT as Stock Points (S).
- s. Cabinets: 0 - 16 for computed; 0 - 8 for extra. The system computes the required numbers for the 4 types of cabinets and presents this in the COMP field. Additional quantities may be entered in the XTRA field within the allowed ranges specified above as desired.
- t. Max Distance Between Computers: A - F. Input is only allowed for sites designated in file CONFIG.SIT as Stock Points (S).

On completion of the Max Distance input value, confirm the input values by entering a "Y" to the prompt " Do you accept the input values thus far? Yes or No ". The default value is "N".

Screen 6: Select various software packages and the number of both NETEX and SPLICENet software maintenance months desired. The system only accepts "Y" or "N" entries for software packages. The system only accepts integers in the range 0 - 12 for software maintenance months entries. Network Maintenance Facility (NMF) software is divided into either a group package or individual packages. If the user selects the group package, none of the individual packages can be selected. The cursor moves directly to the NETEX Maintenance Months field. If the NMF group package field response is "N", the user may select each individual package if desired. On completion of the entry for the number of months of SPLICENet software maintenance desired, confirm the input values by entering a "Y" to the prompt " Do you

accept the input values thus far? Yes or No ".
The default value is "N".

Screen 7: Enter the quantities for system documentation, training group and courses, and months of Emergency Per-Call Maintenance. Indicate whether to include Site Preparation charges.

The allowable range for documentation and training courses is 0 - 20. The allowable range for Training Groups is 1 - 5. The allowable range for months of Emergency Maintenance is 0 - 12. The allowable inputs to Site Prep charges are "Y" or "N". On completion of the Site Prep charges, confirm the input values by entering a "Y" to the prompt " Do you accept the input values thus far? Yes or No ". The default value is "N".

Screen 8: The configurer software module sign-off screen requires no input. The system displays the output file name used for this configuration run in the sign-off message.

The system returns to the Function Selection Menu (Screen 1) to await the next selection.

2.3.2 FUNCTION 2: Perform LOTUS 1-2-3 Financial or "What-If" Analysis

Discussion of the following actions is predicated on the user having a well developed understanding of the LOTUS 1-2-3 system. Terminate the system and review any of several available books detailing the system's capabilities and operations before continuing if you are not familiar with that software product.

Select option 2 to begin LOTUS 1-2-3 financial or "what-if" analysis processing. Insert a LOTUS system disk in drive A (or have a product such as ZERODISK⁴ installed) to start the LOTUS system. If the Function Selection Menu is not displayed, select the "Return to " option of the current menu until the Function Selection Menu appears. If

⁴ ZERODISK is a software product available from Quaid Software Limited, 45 Charles Street East, Third Floor, Toronto, Ontario M4Y 1S2 (416) 961-8243. It is a product that enables users to run software applications without the need to place master disks in the "A" drive required by some programs such as dBASE III, LOTUS 1-2-3, etc.

a process is active, select the option that terminates the process. Once a menu appears, select the "Return to " option of the current menu until the Function Selection Menu appears. Insert a LOTUS system disk in drive A and then select option 2 when the Function Selection Menu appears.

Following a message concerning changing the LOTUS active file directory, the first screen of the LOTUS 1-2-3 system (Screen 9 - See Attachment 2 for screen formats) appears. The system experiences a few seconds delay for system startup.

NOTE: a backslash (\) followed by a single letter indicates a LOTUS macro. Execute a macro by simultaneously depressing the ALT and letter keys. A slash (/) followed by a letter indicates a LOTUS command. [CR] denotes the striking of the RETURN or ENTER key.

Screen 9: The opening menu of the LOTUS 1-2-3 system requires no input. Processing continues with the depression of any key.

Screen 10: The empty LOTUS 1-2-3 spreadsheet screen appears. Change the default subdirectory in LOTUS if it is not subdirectory C:\DBASEIII. Enter LOTUS command /WGDDC:\DBASEIII[CR]Q to change the default subdirectory. Enter LOTUS command /FR to retrieve a file. Screen 11 appears. Use the arrow keys to point to SKELETON or MAINTORD or type either SKELETON or MAINTORD. SKELETON.WKS is the formatting file for outputs from the Pascal Configurer module. This file includes the macros developed for recalculation analysis beginning in cell A200. MAINTORD is the formatting file for outputs from the Maintenance Delivery Order Generation module executed from within the dBASE Configuration Management System. This file includes macros similar to those beginning in cell A200 of file SKELETON.WKS. If the user selects the SKELETON worksheet, Screen 12 - the formatted spreadsheet, appears.

Screen 13: Enter the LOTUS command /FIN{file name} or the macro \F{file name} to begin the importation process. Enter an output file name generated by the Pascal Configurer module. It may either be typed in without the ".PRN" extension or selected by pointing to the file name with the arrow keys.

No further screens for the LOTUS processes are shown here. All screens appear the same, showing different views of the memory resident spreadsheet.

The following LOTUS macros in file SKELETON.WKS have been provided for easier processing:

- a. \C - Changes column numeric entries to currency. Execute the macro anywhere in the worksheet.
- b. \D - Deletes indicated rows. Place the cursor at the first row to delete before entering \D. Point to the last row to delete using the arrow keys.
- c. \E - Deletes all ".PRN" files. Execute the macro anywhere in the worksheet.
- d. \F - Imports a ".PRN" file at the cursor position. Execute the macro anywhere in the worksheet.
- e. \I - Recalculates the Total Component Installation Price for a row. Place the cursor in the top row cell of the newly created temporary column (e.g., hardware, software, etc.). Copy subsequent entries using /C versus using \I.
- f. \M - Recalculates the Total Component Purchase Price for a row. Place the cursor in the top row cell of the newly created temporary column (e.g., hardware, software, etc.). Copy subsequent entries using /C versus using \M.
- g. \N - Recalculates the Total Hardware Component Maintenance Price for a row. Place the cursor in the top row cell of the newly created temporary column (e.g., hardware, software, etc.). Copy subsequent entries using /C versus using \N.
- h. \O - Recalculates the Total Software Component Maintenance Price for a row. Place the cursor in the top row cell in the newly created temporary column (e.g., hardware, software, etc.). Copy subsequent entries using /C versus using \O.
- i. \P - Prepares the worksheet for output to the dBASE process. Execute the macro anywhere in the worksheet.

- j. \R - Names a macro. Execute the macro in the cell of the new macro identifier.
- k. \S - Sum indicated columns. Execute the macro from the cell where the total figure is desired. Use arrow keys, followed by the RETURN or ENTER key, to indicate the beginning and end of the summary area.
- l. \T - Recalculates the Component Downtime hourly rate. Place the cursor in the top row cell in the newly created temporary column (e.g., hardware, software, etc.). Copy subsequent entries using /C versus using \T.
- m. \U - Recalculates the Component System Downtime hourly rate. Place the cursor in the top row cell in the newly created temporary column (e.g., hardware, software, etc.). Copy subsequent entries using /C versus using \U.

Perform formula recalculation one column at a time starting from the left. Insert a new column to the left of the Total Purchase Price, Total Component Maintenance, Total Installation Price, and two Downtime Credit columns. Execute the \M, \N, \O, \I, \T, and \U macros described above in the first entry of each applicable column. Copy the resulting formula down the remainder of the column. Sum the column using the \S macro. When results are satisfactory, move (/M) the new column over the old column and delete (/WDC) the now blank column. Re-sum (no macro provided) the summary financial data at the bottom of the spreadsheet.

Perform "what-if" analysis, using the macros provided, following formula recalculation. **Exercise extreme care when changing component quantities!** If component quantity changes are made, print and review the proposed changes. After reviewing the changes, reverify the accuracy of the changes using the Configurer system. Use the Configurer to ensure that all configuration rules are properly followed.

Save an archival copy of the worksheet with the /FS{file name} command. Print a delivery order with the /PP command. Strip off the worksheet headers, non-hardware and software line items, section cost totals, summary notes and cost information with the \D macro. Print the remaining contents of the spreadsheet (less macros) with the /PF{file name} command or \P macro.

Terminate 1-2-3 by entering the LOTUS command /QY[CR]. The system returns to the Function Selection Menu (Screen 1) to await the next selection.

The following processing is accomplished if the file MAINTORD is selected. The system automatically loads the NEWDO.PRN file created from the dBASE III Maintenance Delivery Order Generation module. The cursor moves to the appropriate field to accept entry of the effective date. Use the macros stored at location A200 to verify and complete the maintenance delivery order following entry of the effective date.

The following LOTUS macros on MAINTORD.WKS have been provided for easier processing:

- a. \C - Copies header information.
- b. \D - Deletes the first column.
- c. \O - Automatically imports the maintenance delivery order called NEWDO.PRN.
- d. \I - Adds rows for software headers.
- e. \N - Recalculates the Total Hardware Component Maintenance Price for a row. Place the cursor in the top row cell in the newly created temporary column (i.e., hardware and software). Copy subsequent entries using /C versus using \N.
- f. \O - Recalculates the Total Software Component Maintenance Price for a row. Place the cursor in the top row cell in the newly created temporary column (i.e., hardware and software, etc.). Copy subsequent entries using /C versus using \O.
- g. \R - Names a macro. Execute the macro in the cell of the new macro identifier.
- h. \S - Sum indicated columns. Execute the macro from the cell where the total figure is desired. Use arrow keys, followed by the RETURN or ENTER key, to indicate the beginning and end of the summary area.

Locate the first software item in the body of the spreadsheet (feature number between 510101 and 660101, 860101 or 860201). Move the cursor to the corresponding location in column "A". Execute the \I macro to insert blank rows at the location. When complete, move the cursor down 4 rows and execute the /C LOTUS command to copy headers to the beginning of the next section.

Verify the calculated Component Factored Maintenance cell for each data entry. Move the cursor to the first entry in the hardware section of the Component Factored Maint column and execute the command /WIC[CR]. This will add an additional column to the spreadsheet. Execute macro \N to automatically recalculate the maintenance amount at the first hardware component cell. Execute the LOTUS command /C[CR]{DOWN}.{DOWN to the end of the hardware column}[CR]. This copies the formula in the first cell to all following cells. Use the \S macro to sum the column and copy the same formula to the next cell to the right with the /C LOTUS command.

Comparison of these two sums may show minor rounding differences. Use the /M command to move the desired cells one column to the right to retain the LOTUS figure. Use the same procedure in the software section, substituting the \O macro for the \N macro. Delete the unnecessary column with the /WDC command following the movement of the data to the newly created column.

When validation of all entries is complete, manually enter financial appropriation data and end of delivery order comments. Manually recalculate a new System Downtime Credit Factor value using data supplied on the spreadsheet plus the installation cost. Save or print the new delivery order, as desired.

Terminate LOTUS 1-2-3 by executing the LOTUS command /QY[CR]. The system returns to the Function Selection Menu (Screen 1) to await the next selection.

2.3.3 FUNCTION 3: Execute the dBASE III Configuration Management System

Select menu option 3 (from the Function Selection Menu - Screen 1) to invoke the dBASE III Configuration Management System. If the Function Selection Menu is not displayed, select the "Return to " option of the current menu until the Function Selection Menu appears. If a

process is active, select the option that terminates the process. Once a menu appears, select the "Return to " option of the current menu until the Function Selection Menu appears. Next select menu option 3. The first screen of the dBASE III Configuration Management System (the Process Selection Menu - Screen 14) appears.

Screen 14: Ten menu options (0 - 9) exist. Processing continues based on the selection entered. Option 0 returns the system to the Function Selection Menu (Screen 1). The remaining options are discussed in order.

2.3.3.1 Load a new Delivery Order into the Configuration Management System.

Select menu option 1 (from the Function Selection Menu - Screen 1) to load a new delivery order generated by the SPLICE Configurer. The Delivery Order Load Menu (Screen 15) appears. Next select menu option 1 to commence the loading process for the new delivery order.

Screen 15: Select one of two options: 1 - load a new delivery order or 2 - return to the Process Selection Menu (Screen 14).

Screen 16: Enter the LOTUS output file name. A file name may be from one to eight alphanumeric characters long. The default file name supplied by the system is "SPLICE.PRN". The system automatically provides the extension. If the file name entered cannot be found on the default subdirectory, re-enter a valid name. An error message appears on the status line if the file name entered cannot be found. After three invalid entries, either exit the program or supply another file name. When a valid file name is supplied, enter the effective date for the delivery order.

Valid dates range from 840101 to 991231 (the system **currently will not accept leap year dates - 29 February**). The actual site number from the input delivery order appears following the entry of a valid date. The user may change the site number to any site number within the range 01 - 58 or accept the site number displayed. Following the entry of a valid site number, accept all data entries before the load process begins. If the response is "N", all data entries are erased and the input process is repeated. If the response is "Y", indicate input file disposition: 1 - retain or 2 - erase.

The update process commences following this response. The load process may take up to 10 minutes. This is primarily due to the building of serial number records for each individual component on the delivery order. **BE PATIENT.** During the load process, status messages appear to keep the user apprised of the transactions as they occur. When the load process finishes, indicate whether to load another delivery order. If the response is "Y", the process starts with a new Screen 16. If the response is "N", the system returns to the Delivery Order Load Menu (Screen 15). Select menu option 2 to return to the Process Selection Menu (Screen 14) to await the next selection.

2.3.3.2 Perform maintenance on the Equipment File.

Select menu option 2 (from the Process Selection Menu - Screen 14) to either modify or review records in the Equipment File. Following the selection of option 2, the Equipment Maintenance Selection Menu (Screen 17) appears.

Screen 17: The Equipment Maintenance Selection Menu enables the user to review or modify selected entries in the Equipment File. Select one of three options: 1 - update price information; 2 - review equipment file entries; or 3 - return to the Process Selection Menu (Screen 14).

2.3.3.2.1 Modify an Equipment File Record.

Select menu option 1 (from the Equipment Maintenance Selection Menu - Screen 17) to modify an Equipment File record. Following the selection of option 1, the Equipment Update Format screen (Screen 18) appears.

Screen 18: Enter the site number to update, an integer from 01 to 58.

Select one of three options: 1 - update a specific site's records; or 2 - start at the beginning of the Equipment File; or 3 - start at the end of the Equipment File viewing records until the desired record appears. Enter "00" to start viewing records at the beginning of the file. Enter "99" to start viewing records at the end of the file. Enter the site number to modify records for a specific site.

Following entry of the specified site number, two options exist: 1 - select a specific six digit feature number or 2 - start with the first feature number by entering "00 " (two zeroes followed by four spaces). Status messages appear at the bottom of the screen when reaching the first and last records. This action only occurs if a specific site is selected (a site selection other than "00" or "99").

The only authorized changes in this screen are the three price fields. An introductory window, explaining how to terminate the modification of a record field, appears following the entry of a feature number. Terminate the introductory information window by striking the RETURN or ENTER key. Changes to fields are possible one field at a time. If changes are made to any field, either accept or reject the changes. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the update process. On exiting, the system returns to the Equipment Maintenance Selection Menu (Screen 17).

2.3.3.2.2 Review an Equipment File Record.

Select menu option 2 (from the Equipment Maintenance Selection Menu - Screen 17) to review an Equipment File record. Following the entry of option 2, the Equipment Review Format screen (Screen 19) appears.

Screen 19: Enter the site number to review, an integer from 01 to 58.

Select one of three options: 1 - review a specific site's records; or 2 - start at the beginning of the Equipment File; or 3 - start at the end of the Equipment File viewing records until the desired record appears. Enter "00" to start viewing records at the beginning of the file. Enter "99" to start viewing records at the end of the file. Enter the site number to review records for a specific site.

Following the entry of specified site number, two options exist: 1 - select a specific six digit feature number or 2 - start with the first feature number by entering "00 " (two zeroes followed by four spaces). Status messages appear at the bottom of the screen when reaching the first and last records. This action only occurs if a specific site is selected (a site selection other than "00" or "99").

No changes are allowed in this screen. Choose either:
1 - access the next record; 2 - access the previous record;
or 3 - exit the review process. On exiting, the system
returns to the Equipment Maintenance Selection Menu
(Screen 17).

2.3.3.3 Perform Maintenance on the Equipment Description File.

Select menu option 3 (from the Process Selection Menu - Screen 14) to either modify or review records in the Equipment Description File. Following the selection of option 3, the Equipment Description Maintenance Menu (Screen 20) appears.

Screen 20: The Equipment Description Maintenance Menu enables the user to review or modify selected entries in the Equipment Description File. Select one of three options:
1 - modify Equipment Description File entries; 2 - review Equipment Description File entries; or 3 - return to the Process Selection Menu (Screen 14).

2.3.3.3.1 Modify an Equipment Description File Record.

Select menu option 1 (from the Equipment Maintenance Selection Menu - Screen 17) to modify an Equipment Description File record. After the selection of option 1, the Description Update Format screen (Screen 21) appears.

Screen 21: Enter: 1 - "00 " (two zeroes followed by four spaces) to start the update process at the top of the file; 2 - "99 " (two nines followed by four spaces) to start at the update process the end of the file; or 3 - a six digit feature number. Valid feature numbers range from 000101 to 994001.

An introductory window, explaining how to terminate the modification of a record field, appears following the entry of a feature number. Terminate the introductory information window by striking the RETURN or ENTER key. Changes to fields are possible one field at a time.

All data entries in this screen may be modified. Once the Base Maintenance Price field is either modified or passed, the user may update the memo field. If the response is "Y", a window of instructions (Screen 22) appears. The

instructions describe how to make changes to the memo field. If the response is "N", processing continues.

Accept or reject changes made to any field. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the update process. On exiting, the system returns to the Equipment Maintenance Selection Menu (Screen 17).

2.3.3.3.2 Review an Equipment Description File Record.

Select menu option 2 (from the Equipment Maintenance Selection Menu - Screen 17) to review an Equipment Description File record. After the selection of option 2, the Description Review Format screen (Screen 23) appears.

Screen 23: Enter either: 1 - "00 " (two zeroes followed by four spaces) to start the update process at the top of the file; 2 - "99 " (two nines followed by four spaces) to start at the update process the end of the file; or 3 - a six digit feature number. Valid feature numbers range from 000101 to 994001.

No changes are allowed in this screen. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the review process. On exiting, the system returns to the Equipment Maintenance Selection Menu (Screen 17).

2.3.3.4 Perform Maintenance on the Site Name File.

Select menu option 4 (from the Process Selection Menu - Screen 14) to either modify or review records in the Site Name File. Following the selection of option 4, the Site Name Maintenance Menu (Screen 24) appears.

Screen 24: The Site Name Maintenance Menu enables the user to review or modify selected entries in the Site Name File. Select one of three options: 1 - modify Site Name File entries; 2 - review Site Name File entries; or 3 - return to the Process Selection Menu (Screen 14).

2.3.3.4.1 Modify a Site Name File Record.

Select menu option 1 (from the Site Name Maintenance Menu - Screen 24) to modify a Site Name File record. After the selection of option 1, the Site Address Data Update Format screen (Screen 25) appears.

Screen 25: Enter the site number to update, an integer from 01 to 58.

Select one of three options: 1 - update a specific site's records; or 2 - start at the beginning of the Site Name File; or 3 - start at the end of the Site Name File viewing records until the desired record appears. Enter "00" to start viewing records at the beginning of the file. Enter "99" to start viewing records at the end of the file. Enter the site number to modify records for a specific site.

All data entries, except site number and type activity, may be changed. An introductory window, explaining how to terminate the modification of a record field, appears following the entry of a feature number. Terminate the introductory information window by striking the RETURN or ENTER key. Changes to fields are possible one field at a time. Accept or reject changes made to any field. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the update process. On exiting, the system returns to the Site Name Maintenance Menu (Screen 24).

2.3.3.4.2 Review a Site Name File Record.

Select menu option 2 (from the Site Name Maintenance Menu - Screen 24) to review a Site Name File record. Following the selection of option 2, the Site Address Data Review Format screen (Screen 26) appears.

Screen 26: Enter the site number to review, an integer from 01 to 58.

Select one of three options: 1 - review a specific site's records; or 2 - start at the beginning of the Site Name File; or 3 - start at the end of the Site Name File viewing records until the desired record appears. Enter "00" to start viewing records at the beginning of the file.

Enter "99" to start viewing records at the end of the file.
Enter the site number to review records for a specific site.

No data entries may be changed in this screen. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the update process. On exiting, the system returns to the Site Name Maintenance Menu (Screen 24).

2.3.3.5 Perform Maintenance on the Manual File.

Select menu option 5 (from the Process Selection Menu - Screen 14) to either modify or review records in the Manual File. Following the selection of option 5, the Manual Maintenance Menu (Screen 27) appears.

Screen 27: The Manual Maintenance Menu enables the user to either access, modify, add or delete selected entries in the Manual File. Select one of five options: 1 - add a new Manual Description entry; 2 - update Manual Description entries; 3 - delete a Manual Description entry; 4 - review Manual Description entries; or 5 - return to the Process Selection Menu (Screen 14).

2.3.3.5.1 Add a new Manual Description entry.

Manual description entries may only be added for the site selected. The site number and feature number must be known to successfully execute this process. This restriction applies even if a manual description already exists for a site and feature number. Be sure you want to add a new manual and not just update an existing one! Delete an old manual if it is no longer applicable.

Screen 28: Enter a valid site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site number selected.

Enter the feature number for the manual description to add. Valid feature numbers range from 000101 to 994001. The system validates the feature number to ensure that the feature number exists on the file. Once a valid feature number is entered, the CLIN and description data appear. The cursor moves to the Manual Description field where the new manual description is entered. Indicate whether the new description is acceptable. If the response is "N", either choose to continue or exit. If the response is "Y", the new

description entered is accepted. Choose either to continue or exit. On exiting, the system returns to the Manual Maintenance Menu (Screen 27).

2.3.3.5.2 Update a Manual Description entry.

Select menu option 2 (from the Manual Maintenance Menu - Screen 27) to modify a Manual File record. After the selection of option 2, the Manual Update Format screen (Screen 29) appears.

Screen 29: Enter a valid site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Select one of three options: 1 - update a specific site's records; or 2 - start at the beginning of the Manual File; or 3 - start at the end of the Manual File viewing records until the desired record appears. Enter "00" to start viewing records at the beginning of the file. Enter "99" to start viewing records at the end of the file. Enter the site number to modify records for a specific site.

Following entry of the specified site number, two options exist: 1 - select a specific six digit feature number or 2 - start with the first feature number by entering a feature number of "00 " (**two zeroes followed by four spaces**). Status messages appear at the bottom of the screen when reaching the first and last records. This action only occurs if a specific site is selected (a site selection other than "00" or "99").

The only field allowed to be modified during this process is the Manual Description field. An introductory window, explaining how to terminate the modification of a record field, appears following the entry of a feature number. Terminate the introductory information window by striking the RETURN or ENTER key.

Changes to fields are possible one field at a time. Accept or reject changes made to any field. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the update process. On exiting, the system returns to the Manual Maintenance Menu (Screen 27).

2.3.3.5.3 Delete a Manual Description entry.

Select menu option 3 (from the Manual Maintenance Menu - Screen 27) to delete a Manual Description entry. After the selection of option 3, the Manual Deletion Format screen (Screen 30) appears.

Screen 30: Enter a valid site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Enter the feature number for the description to delete. Valid feature numbers range from 000101 to 994001. When the description appears, verify the deletion decision. If the response is "N", the Manual Description is left intact. If the response is "Y", the Manual Description is deleted. Choose either to continue or exit. On exiting, the system returns to the Manual Maintenance Menu (Screen 27).

2.3.3.5.4 Review a Manual Description entry.

Select menu option 4 (from the Manual Maintenance Menu - Screen 27) to review a Manual Description entry. After the selection of option 4, the Manual Review Format screen (Screen 31) appears.

Screen 31: Enter a site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Select one of three options: 1 - review a specific site's records; or 2 - start at the beginning of the Manual File; or 3 - start at the end of the Manual File viewing records until the desired record appears. Enter "00" to start viewing records at the beginning of the file. Enter "99" to start viewing records at the end of the file. Enter the site number to review records for a specific site.

No data entries may be changed in this screen. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the update process. On exiting, the system returns to the Site Name Maintenance Menu (Screen 24).

2.3.3.6 Perform Maintenance on the Serial Number File.

Select menu option 6 (from the Process Selection Menu - Screen 14) to either modify or review records in the Serial Number File. Following the selection of option 6, the Serial Number Maintenance Menu (Screen 32) appears.

Three data elements must be known to perform an update on a Serial Number File record. The three data elements are: 1 - site number, 2 - effective delivery order date and 3 - feature number of the serial number to be modified. If **all three or any** of these data elements are not known, run a date level report to obtain the three elements (refer to the section Generate REPORTS for the Project, a Site or Equipment for specific procedures).

2.3.3.6.1 Modify a Serial Number File record.

Select menu option 1 (from the Serial Number Maintenance Selection Menu - Screen 32) to modify a Serial Number File record. After the selection of option 1, the Serial Number Update Format screen (Screen 33) appears.

Screen 33: Enter a valid site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Following the site number entry, enter an effective delivery order date. Three attempts are allowed to specify an effective delivery order date. Screen 34 appears if on the third attempt a valid effective delivery order date is not entered. Select one of two choices: 1 - continue with the update process or 2 - exit the update process and obtain the three elements (refer to the section Generate REPORTS for the Project, a Site or Equipment for specific procedures).

Once a delivery order date is entered, enter a valid feature number. Valid feature numbers range from 000101 to 994001. Screen 34 appears if all three data elements do not match any record data fields for the site selected. The same two choices described in the paragraph above may be chosen. When a valid feature number is entered and all three data elements match, a short introductory window explaining how to terminate the modification of a record

field appears. Terminate the introductory information window by striking the RETURN or ENTER key.

Following termination of the introductory information screen, the Serial Number File record selected appears. The only field that may be modified is the serial number field. Accept or reject changes made to the serial number field. If the response is "Y", the change is made to the database. If the response is "N", the change is not accepted. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the update process. On exiting, the system returns to the Serial Number Maintenance Menu (Screen 32).

2.3.3.6.2 Review a Serial Number File record.

Select menu option 2 (from the Serial Number Maintenance Selection Menu - Screen 32) to review a Serial Number File record. After the selection of option 2, the Serial Number Update Format screen (Screen 35) appears.

Screen 35: Enter a valid site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Select one of three options: 1 - review a specific site's records; or 2 - start at the beginning of the Serial Number File; or 3 - start at the end of the Serial Number File viewing records until the desired record appears. Enter "00" to start viewing records at the beginning of the file. Enter "99" to start viewing records at the end of the file. Enter the site number to review records for a specific site.

Following entry of the specified site number, two options exist: 1 - select a specific six digit feature number or 2 - start with the first feature number by entering "00 " (two zeroes followed by four spaces). Valid feature numbers range from 000101 to 994001. Status messages appear at the bottom of the screen when reaching the first and last records. This action only occurs if a specific site is selected (a site selection other than "00" or "99").

No data fields are allowed to be modified during the review process. Choose either: 1 - access the next record; 2 - access the previous record; or 3 - exit the review

process. On exiting, the system returns to the Serial Number Maintenance Menu (Screen 32).

2.3.3.7 Generate REPORTS for the Project, a Specific Site or a Particular Date.

Select menu option 7 (from the Process Selection Menu - Screen 14) to obtain an overall project report, a report for a particular site or a report for a delivery order issued on a particular site. Following the selection of option 7, the Report by Type Menu (Screen 36) appears.

Screen 36: Various levels of reports which may be selected appear. Select one of three options: 1 - obtain a project level report; 2 - obtain a site specific report; 3 - obtain a delivery order specific report; or 4 - return to the Process Selection Menu (Screen 14).

Screen 37: When obtaining any of the various types of reports, two options exist: 1 - obtain a printed report or 2 - view the data on screen. Screen 37 always appears if a printed report is selected. Ensure: 1 - the power to the printer is on; 2 - sufficient paper is loaded in the printer and 3 - the leading edge of the paper is positioned with the printer's typing line alignment mark. After all three conditions are satisfied, commence printing by the striking the RETURN or ENTER key. Once printing commences, the appropriate screen appears and status messages detailing the progress of the report are displayed.

2.3.3.7.1 Obtain an Overall Project Level Report.

Select menu option 7 (from the Process Selection Menu - Screen 14) to obtain an overall project level report for a site. The Report by Type Menu (Screen 36) appears. From the Report by Type Menu, select option 1. After the selection of option 1, the Project Level Reports Menu (Screen 38) appears.

Screen 38: Select one of three options: 1 - obtain a report by equipment type; 2 - obtain a report by serial numbers; or 3 - return to the Report by Type Menu (Screen 36).

2.3.3.7.1.1 Obtain an Overall Project Report by Equipment Type.

Select menu option 1 (from the Project Level Reports Menu - Screen 38) to obtain an overall project report broken down by type of equipment. After the selection of option 1, the Equipment Project Level Report screen (Screen 39) appears.

Screen 39: Indicate whether a printed report is desired. Responses are either "Y" or "N". If the response is "Y", Screen 37 appears before printing commences (See the discussion of Screen 37 above). If the response is "N", the data and associated headings appear. Screen 40 is a sample report format.

Screen 40: All equipment is totaled by feature number and presented. The quantity for each feature number displayed represents the total quantity ordered for all sites in the Equipment database. After each screen, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Project Level Reports Menu (Screen 38).

2.3.3.7.1.2 Obtain an Overall Project Report by Serial Number.

Select menu option 2 (from the Project Level Reports Menu - Screen 38) to obtain an overall project report broken down by serial number. After the selection of option 2, the Equipment Serial Number Project Level Report screen (Screen 41) appears.

Screen 41: Indicate whether a printed report is desired. Responses are either "Y" or "N". If the response is "Y", Screen 37 appears before printing commences (See the discussion of Screen 37 above). If the response is "N", the data and associated headings appear. Screen 42 is a sample report format.

Screen 42: All serial numbers for each component at all sites are presented. This will probably be a LARGE report! Entries include: Site Number, CLIN, Feature Number, Description, Effective Delivery Order Date, total component quantity on the delivery order, specific component number (e.g. 1 of 9), and the applicable serial number. After each

screen, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Project Level Reports Menu (Screen 38).

2.3.3.7.2 Obtain a Report for a Particular Site.

Select menu option 7 (from the Process Selection Menu - Screen 14) to obtain a report for a particular site. The Report by Type Menu (Screen 36) appears. Select menu option 2 from the Report by Type Menu. After the selection of option 2, the Site Level Reports Menu (Screen 43) appears.

Screen 43: Select one of four options: 1 - obtain a report by equipment type; 2 - obtain a report of site manuals; 3 - obtain a report by serial number; or 4 - return to the Site Level Reports Menu (Screen 43).

2.3.3.7.2.1 Obtain a Site Specific Report by Equipment Type.

Select menu option 1 (from the Site Level Reports Menu - Screen 43) to obtain a site specific report broken down by equipment type. After the selection of option 1, the Equipment Site Level Report screen (Screen 44) appears.

Screen 44: Enter the site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Screen 45: Indicate whether a printed report is desired. Responses are either "Y" or "N". If the response is "Y", Screen 37 appears before printing commences (See the discussion of Screen 37 above). If the response is "N", the data and associated headings appear. Screen 46 is a sample report format.

Screen 46: All records for a specific site are selected from the Equipment database and their quantities are totaled. The Site Number, CLIN, Feature Number, Equipment Description, and total site quantity are presented. After each screen, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Site Level Reports Menu (Screen 42).

2.3.3.7.2.2 Obtain a Site Specific Report of Manuals.

Select menu option 2 (from the Site Level Reports Menu - Screen 43) to obtain a site specific manual report. After the selection of option 2, the Site Level Manual Report screen (Screen 47) appears.

Screen 47: Enter the site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Screen 48: Indicate whether a printed report is desired. Responses are either "Y" or "N". If the response is "Y", Screen 37 appears before printing commences (See the discussion of Screen 37 above). If the response is "N", the data and associated headings appear. Screen 49 is a sample report format.

Screen 49: The Manual File is accessed and each feature number within the selected site appears. Report items include Site Number, CLIN, Feature Number, Description, and Manual Description. After each screen, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Site Level Reports Menu (Screen 42).

2.3.3.7.2.3 Obtain a Site Specific Report by Serial Number.

Select menu option 3 (from the Site Level Reports Menu - Screen 43) to obtain a site specific report of serial numbers. After the selection of option 3, the Site Serial Number Report screen (Screen 50) appears.

Screen 50: Enter the site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Screen 51: Indicate whether a printed report is desired. Responses are either "Y" or "N". If the response is "Y", Screen 37 appears before printing commences (See the discussion of Screen 37 above). If the response is "N", the data and associated headings appear. Screen 52 is a sample report format.

Screen 52: All serial numbers for each component at a site appear. Entries include: Site Number, CLIN, Feature Number, Description, Effective Delivery Order Date, total component quantity on the delivery order, specific component number (e.g. 1 of 9), and the applicable serial number. After each screen, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Site Level Reports Menu (Screen 42).

2.3.3.7.3 Obtain a Report for a Delivery Order Issued on a Particular Date.

Select menu option 7 (from the Process Selection Menu - Screen 14) to obtain a report for a delivery order issued on a particular date. The Report by Type Menu (Screen 36) appears. From the Report by Type Menu, select option 3. After the selection of option 3, the Delivery Order Date Level Reports Menu (Screen 53) appears.

Screen 53: Select one of four options: 1 - obtain an equipment report with unit costs; 2 - obtain an equipment report without costs; 3 - obtain a report by serial number; or 4 - return to the Delivery Order Date Level Reports Menu (Screen 53).

2.3.3.7.3.1 Obtain a Report by Equipment Type with Unit Prices.

Select menu option 1 (from the Delivery Order Date Level Reports Menu - Screen 53) to obtain a date level report broken down by equipment type with unit prices. After the selection of option 1, the Delivery Order Level Report screen (Screen 54) appears.

Screen 54: Enter the site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Screen 55: All delivery orders existing in the Equipment database appear. After the last delivery order effective date appears, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Delivery Order Date Level Reports Menu (Screen 53).

Screen 56: Indicate whether a printed report is desired. Responses are either "Y" or "N". If the response

is "Y", Screen 37 appears before printing commences (See the discussion of Screen 37 above). If the response is "N", the data and associated headings appear. Screen 57 is a sample report format.

Screen 57: Items appearing on the report are Delivery Order Date, Site Number, CLIN, Feature Number, Description, Delivery Order Quantity, and Component Unit Purchase Price. After each screen, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Delivery Order Date Level Reports Menu (Screen 53).

2.3.3.7.3.2 Obtain a Report by Equipment Type without Unit Prices.

Select menu option 2 (from the Delivery Order Date Level Reports Menu - Screen 53) to obtain a date level report broken down by equipment type without unit prices. After the selection of option 2, the Delivery Order Level Report screen (Screen 58) appears.

Screen 58: Enter the site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Screen 59: All delivery orders existing in the Equipment database appear. After the last delivery order effective date appears, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Delivery Order Date Level Reports Menu (Screen 53).

Screen 60: Indicate whether a printed report is desired. Responses are either "Y" or "N". If the response is "Y", Screen 37 appears before printing commences (See the discussion of Screen 37 above). If the response is "N", the data and associated headings appear. Screen 61 is a sample report format.

Screen 61: Items appearing on the report are Delivery Order Date, Site Number, CLIN, Feature Number, Description, Delivery Order Quantity, and FDC Model Number. After each screen, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Delivery Order Date Level Reports Menu (Screen 53).

2.3.3.7.3.3 Obtain a Date Level Report by Serial Number.

Select menu option 3 (from the Delivery Order Date Level Reports Menu - (Screen 53)) to obtain a date level report broken down by serial number. After the selection of option 3, the Site Serial Number Report screen (Screen 62) appears.

Screen 62: Enter the site number, an integer from 01 to 58. The site number entered is validated to ensure that records exist for the site selected.

Screen 63: All delivery orders existing in the Equipment database appear. After the last delivery order effective date appears, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Delivery Order Date Level Reports Menu (Screen 53).

Screen 64: Indicate whether a printed report is desired. Responses are either "Y" or "N". If the response is "Y", Screen 37 appears before printing commences (See the discussion of Screen 37 above). If the response is "N", the data and associated headings appear. Screen 65 is a sample report format.

Screen 65: Items appearing on the report are Delivery Order Date, Site Number, CLIN, Feature Number, Description, Delivery Order Effective Date, Total Quantity by Component ordered on the delivery order, specific component quantity (e.g. 1 of 9), and Item Serial Number. After each screen, choose either: 1 - continue the report or 2 - exit the report process. On exiting, the system returns to the Delivery Order Date Level Reports Menu (Screen 53).

2.3.3.8 Generate a Maintenance Delivery Order for a SPLICE Site.

Select menu option 8 (from the Process Selection Menu - Screen 14) to generate a maintenance delivery order for a SPLICE site. Following the selection of option 8, the Maintenance Delivery Order Generation Program screen (Screen 66) appears.

Screen 66: Enter the following data: 1 - Site Number; 2 - LCN Hardware Maintenance Escalation Rate; 3 - LCN

Software Escalation Maintenance Rate; 4 - SPLICENet Maintenance Discount Rate; and 5 - Site Maintenance Escalation Rate. After these data elements are entered, choose either: 1 - continue or 2 - exit the process. If the response is "Y", the maintenance delivery order generation process is initiated and takes approximately 10 minutes to complete. **The output file generated is always "NEWDO.PRN".** On completion of the generation process, the system returns to the Process Selection Menu (Screen 14) to await the next selection.

On returning to the Process Selection Menu, select menu option 0 to return to the Function Selection Menu (Screen 1). From the Function Selection Menu, select menu option 2. After selecting option 2, the system transfers to the LOTUS 1-2-3 environment. Refer to section 2.3.2 on page 19 to obtain the specific details for step-by-step procedures. Since this is a maintenance delivery order rather than an initial delivery order, follow the procedures which address MAINTDO worksheet execution versus SKELETON worksheet execution.

2.3.3.9 Generate Mailing Labels for all SPLICE Sites.

Select menu option 9 (from the Process Selection Menu - Screen 14) to generate mailing labels for all SPLICE sites. Following the selection of option 9, the Mailing Label Generation Program screen (Screen 67) appears.

Screen 67: The mailing label generation program simply produces mailing labels for all the SPLICE sites. Delivery order changes, contract amendments, or other SPLICE related correspondence may be mailed to all SPLICE sites without having to manually create labels. The only input required for the process is the number of copies of mailing labels desired during the run. Valid input values are from 1 to 10 copies of mailing labels. When processing is complete, control returns to the Process Selection Menu (Screen 14) to await the next selection.

This completes the discussion of the process functions of the SPLICE Configurer and dBASE III Configuration Management System. Exit the integrated system by either of two options: 1 - select Function Selection Menu option 5 to return to the dBASE III system prompt or 2 - select Function Selection Menu option 6 to return to the DOS operating system prompt.

AD-A168 517

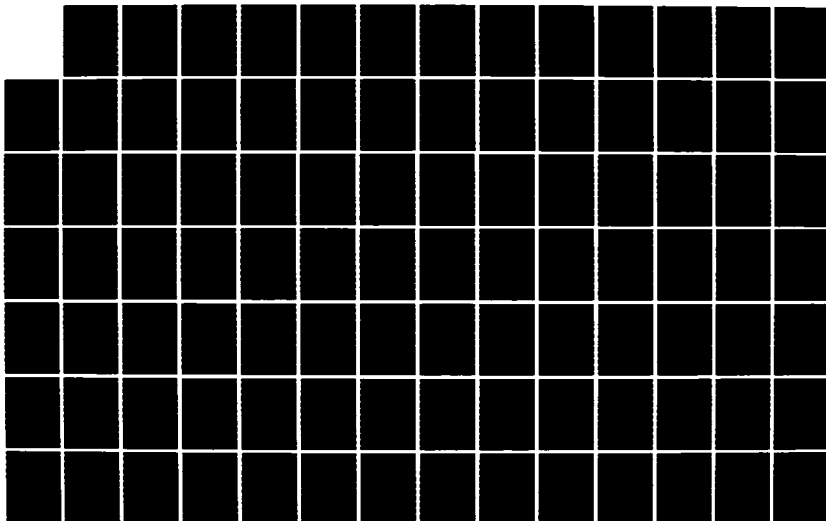
DEVELOPMENT OF AN AUTOMATED MICROCOMPUTER
KNOWLEDGE-BASED INTEGRATED COM. (U) NAVAL POSTGRADUATE
SCHOOL MONTEREY CA R L BEARD MAR 86

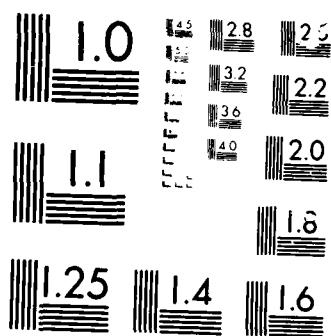
2/5

UNCLASSIFIED

F/O 9/2

ML





Min. Res. (mm)

2.3.4 FUNCTION 4: View the on-line User's Manual

Select Function Selection Menu option 4 to view the on-line User's Manual. The system temporarily transfers control to Wordstar where a copy of the file "USERS.MAN" is viewed. Any changes made to this file during the viewing process are not retained. The file copy is destroyed on termination from Wordstar. Terminate User's Manual viewing by typing either "^KD" or "^KQ" (see note below). Either command returns the system to the Wordstar opening menu. Typing the letter "X" returns the system to the Configuration Management System.

NOTE: The commands "^KD" and "^KQ" are executed by simultaneously holding down the "CTRL" key (represented by the character ^) on the left side of the keyboard and typing the letter "K" followed by either letters "D" or "Q".

3.0 System Output.

The output from the SPLICE Pascal configurer is a formatted disk file. The file is input data for LOTUS 1-2-3, which has 3 outputs: 1 - an archival LOTUS ".WKS" disk file; 2 - a dBASE ".PRN" input disk file; and 3 - a delivery order.

The dBASE process has numerous outputs. Refer to Section 2 (Screens 36 through 66) for further information.

4.0 Exception Reports.

This integrated system is interactive, therefore, no hard copy exception reports are produced. Errorously entered data is presented to the user for immediate action or correction.

5.0 Limitations.

The SPLICE System Configurer was designed on an IBM-PC, but is intended to be run on an IBM-PC AT. The designers recommend that the target IBM-PC AT have the maximum user memory allowed (640KB). To run the dBASE Configuration Management System, a **hard disk is mandatory**. The system requires a 132 column printer to print delivery orders

generated from both LOTUS 1-2-3 and dBASE processes and mailing labels.

If a system other than an IBM-PC/AT is used, the system will respond slowly. Further performance degradation will occur while importing the ".PRN" file into LOTUS. Performance degradation will also occur during the Serial Number building process in the file load and in the maintenance delivery order generation process.

256KB of memory is required if dBASE III version 1.0 is used. 384KB RAM is required if dBASE version 1.1 is used.

The SPLICE Pascal Configurer system is limited by the number of components it can configure (200) and the number of sites it can configure (58).

The LOTUS 1-2-3 and dBASE III modules exhibit only those limitations which exist for those "off-the-shelf" packages.

6.0 Command Sequence.

Issue the command SPLICE (ex: C>SPLICE) from the DOS command prompt to invoke the SPLICE integrated configuration system (Pascal Configurer and dBASE Configuration Management System). This directs DOS to process a command batch file named SPLICE. The command batch file issues all required commands and causes the integrated system to load the memory resident module FLASHUP and commence integrated system execution (See Section 2 for more detailed entries).

NOTE: Prior to issuing the command SPLICE, deactivate any resident color enhancement programs (ex: KOLOR.COM). Such programs interfere with the screen colors generated by the system and data entry color attributes.

7.0 Who to Call.

If program malfunctions occur or questions related to the system arise, contact LCDR E. J. Case, SC, USN, phone number (408) 384-8204 or LCDR R. L. Beard III, SC, USN, phone number (408) 646-1982.

SELECTION CRITERIA FOR A SPLICE CONFIGURATION

SITE NAME: _____

SITE NUMBER: _____

DISCOUNT/ESCALATION RATES:

FDC SNA Interface discount rate: _____

NON-LCN PURCHASE discount rate: _____

LCN PURCHASE discount rate: _____

SPLICENet Software Maintenance discount rate: _____

SPLICENet Software Purchase discount rate: _____

EMERGENCY MAINTENANCE escalation rate: _____

LCN HARDWARE MAINTENANCE escalation rate: _____

LCN SOFTWARE MAINTENANCE escalation rate: _____

INSTALLATION escalation rate: _____

TRAINING escalation rate: _____

DOCUMENTATION escalation rate: _____

MAINTENANCE escalation rate from SPLICE contract: _____

Output File Name: _____ .PRN

Number of MAINTENANCE MONTHS for this order: _____

Effective Delivery Order Date: _____
(MM / DD / YY)

HARDWARE SELECTIONS:

PROCESSORS recommended by FMSO Sizing Study: _____

CENTRONICS PRINTERS to be ordered: _____

TANDEM CRTS to be ordered: _____

128MB DISCs FMSO Sizing Study recommended, **EVEN** No.: _____

240MB DISCs FMSO Sizing Study recommended, **EVEN** No.: _____

540MB DISCs FMSO Sizing Study recommended, **EVEN** No.: _____

Non-6100 ASYNC Controllers to be installed: _____

Non-6100 ASYNC EXTENSION BOARDS to be
installed per controller (0/1/2): _____

BIT SYNC LINES to be supported: _____

BYTE SYNC LINES to be supported: _____

TRI-DENSITY TAPE DRIVES to be installed:
(Ensure fixed disk archival
back-up drives are included) _____

READER/PUNCHES to be installed: _____

CARD READERS to be installed: _____

1000 LPM PRINTERS to be installed: _____

600 LPM PRINTERS to be installed: _____

LCN TRUNKS required for the network: _____

6100 LINE INTERFACE UNITS: _____

PERKIN-ELMER Local Computer Network interfaces: _____

Burroughs B4800 Local Computer Network interfaces: _____

Burroughs B4900 Local Computer Network interfaces: _____

IBM System Local Computer Network interfaces: _____

APPENDIX A: USER's MANUAL

Page A1-47

UNIVAC System Local Computer Network interfaces: _____

FIPS Standard Local Computer Network interfaces: _____

TANDEM HYPERchannels to be installed: _____

PATCH PANEL CABINETS:
(additional for reserve and expansion) _____

SYSTEM CABINETS:
(additional for reserve and expansion) _____

EXPANSION CABINET(S):
(additional for reserve and expansion) _____

HYPERchannel Adapter Cabinet(s) required: _____

Estimate the distance between the two most distant
Computers on the Local Computer Network, Range -
(1 to 5000 feet): _____

SOFTWARE SELECTIONS:

File Security System Software (Yes/No)? _____

LCN File Utility Package Software (Yes/No)? _____

ATP 6100 Software (Yes/No)? _____

BSC 6100 Software (Yes/No)? _____

ADCCP 6100 Software (Yes/No)? _____

BURROUGHS POLL/SELECT 6100 Software (Yes/No)? _____

SNAX and SNAX/HLS 6100 Software (Yes/No)? _____

TINET 6100 Software (Yes/No)? _____

TR 3271 Software (Yes/No)? _____

AM 6520 Software (Yes/No)? _____

T-TEXT Software (Yes/No)? _____

FDC SNA Interface Software (Yes/No)? _____

FDC DLANet Interface Software (Yes/No)? _____

DDN Interface Software (Yes/No)? _____

NETWORK MAINTENANCE FACILITY (NMF):

NMF Group Package Software (Yes/No)? _____

NMF Base Facility Software (Yes/No)? _____

NMF Performance Monitoring Software (Yes/No)? _____

NMF Diagnostic Monitoring Software (Yes/No)? _____

NMF Accounting Application Software (Yes/No)? _____

NETEX MAINTENANCE MONTHS for this order: _____

SPLICENet MAINTENANCE MONTHS for this order: _____

DOCUMENTATION SELECTIONS:

COMPUTER OPERATIONS MANUAL sets required: _____

SYSTEMS PROGRAMMER MANUAL sets required: _____

HARDWARE MANUAL sets required: _____

PROGRAMMER REFERENCE MANUAL sets required: _____

TRAINING SELECTIONS:Select Training Group to be ordered
(Group I-IV / None): _____

OPERATOR TRAINING COURSES required: _____

HARDWARE OVERVIEW COURSES required: _____

SYSTEMS RESOURCE MANAGEMENT COURSES required: _____

SYSTEMS TUNING AND XRAY COURSES required: _____

DATA COMMUNICATIONS COURSES required: _____

TAL COURSES required: _____

SPLICENet Migration Workshop COURSES required: _____

MAINTENANCE AND SITE PREP SELECTIONS:

EMERGENCY PER-CALL MAINTENANCE months required: _____

Should we include SITE PREPS in this run? (Yes/No): _____

FUNCTION SELECTION MENU	
Stock Point Logistics Integrated Communications Environment	
SPLICE	
1	- Configure a site using the SPLICE Configurer
2	- Perform financial analysis of a site using LOTUS 1-2-3
3	- Interact with the Configuration Management System
4	- Review the USER's MANUAL
5	- Return to the dBASE prompt
6	- Return to the DOS Operating System prompt
Please enter your choice: ■	

SCREEN 1

NAVAL SUPPLY SYSTEMS COMMAND		
STOCK POINT LOGISTICS INTEGRATED COMMUNICATIONS ENVIRONMENT		
S P L I C E		
SYSTEM CONFIGURER		
<table border="1"><tr><td>Version 1.1 December 1979</td></tr></table>		Version 1.1 December 1979
Version 1.1 December 1979		

SCREEN 2

LISTING OF SPLICE SITES		
01 ASO PHILADELPHIA	02 FMSO MECHANICSBURG	03 FMSO MECHANICSBURG
04 MCAS CHERRY POINT	05 MCAS EL TORO	06 MCAF QUANTICO
07 MCAS YUMA	08 NAC INDIANAPOLIS	09 NARDAC JACKSONVILLE
10 NARDAC NEW ORLEANS	11 NARDAC NORFOLK	12 NARDAC PENSACOLA
13 NARDAC SAN DIEGO	14 NARDAC SAN FRANCISCO	15 NARDAC WASHINGTON
16 NAS BARBERS POINT	17 NAS BRUNSWICK	18 NAS CECIL FIELD
19 NAS KEY WEST	20 NAEC LAKE HURST	21 NAS MEMPHIS
22 NAS MIRAMAR	23 NAS OCEANA (INACTIVE)	24 NAS PENSACOLA
25 NAS WHIDBEY ISLAND	26 NATC PATUXENT RIVER	27 PMTC POINT MUGU
28 NAVDAF CORPUS CHRISTI	29 NAVDAF GREAT LAKES	30 NAVDAF LEMOORE
31 NAVDAF MOFFETT FIELD	32 NAVDAF ORLANDO	33 NRCC LONG BEACH
34 NRCC NEWPORT	35 NRCC PHILADELPHIA	36 NRCC WASHINGTON
37 NUWES KEYPORT	38 NAVSTA MAYPORT	39 NSC CHARLESTON
40 NSC NORFOLK	41 NSC OAKLAND	42 NSC PEARL HARBOR
43 NSC PUGET SOUND	44 NSC SAN DIEGO	45 NSD GUAM
46 NSD SUBIC BAY	47 NSD YOKOSUKA	48 NSY PHILADELPHIA
49 NSY PORTSMOUTH	50 NTC SAN DIEGO	51 SPCC MECHANICSBURG
52 SUBASE KINGS BAY	53 SUBASE NEW LONDON	54 SUBASE PEARL HARBOR
55 SWFPAC BREMERTON	56 TRF BANGOR	57 SWFLANT KINGS BAY
58 TRF KINGS BAY	59 TO BE DETERMINED	60 TO BE DETERMINED
61 TO BE DETERMINED	62 TO BE DETERMINED	

Please select the site you desire to configure: ■■

SCREEN 3

OUTPUT MEDIA and DISCOUNT/ESCALATION RATES	
DISCOUNT & ESCALATION RATES	
Values input are added to one (1) to generate the correct discount or escalation rate	
FDC SNA Interface Discount Rate:	■■■■
Non-LCN Purchase Discount Rate:	■■■■
LCN Purchase Discount Rate:	■■■■
SPLICENet Software Maintenance Discount Rate:	■■■■
SPLICENet Software Purchase Discount Rate:	■■■■
Emergency Maintenance Escalation Rate:	■■■■
LCN Hardware Maintenance Escalation Rate:	■■■■
LCN Software Maintenance Escalation Rate:	■■■■
Installation Escalation Rate:	■■■■
Training Escalation Rate:	■■■■
Documentation Escalation Rate:	■■■■
Maintenance Escalation Rate:	■■■■
OUTPUT "PRN" FILE NAME	
SPLICE Input and LOTUS 1-2-3 input filename:	■■■■■■■■.PRN
MAINTENANCE MONTHS	
Hardware Maintenance Months: ■■	Delivery Order Effective Date
	Effective Date ■■/■■/■■

SCREEN 4

SYSTEM COMPONENTS	HARDWARE	LCN INTERFACE ADAPTERS
Processors	■ ■ ■	PERKIN-ELMER
Centronics Printers	■ ■ ■	BURROUGHS 4800
Tandem CRTs	■ ■ ■	BURROUGHS 4900
128 MB Disks	■ ■ ■	IBM System
240 MB Disks	■ ■ ■	UNIVAC System
540 MB Disks	■ ■ ■	FIPS Standard
Non-6100 ASYNC Controllers	■ ■ ■	Tandem HYPERchannel
Non-6100 ASYNC Extension Boards	■ ■ ■	
Bit SYNC Lines	■ ■ ■	
Byte SYNC Lines	■ ■ ■	
Tri-Density Tape Drives	■ ■ ■	
Reader/Punches	■ ■ ■	
Card Readers	■ ■ ■	
1000 LPM Printers	■ ■ ■	
600 LPM Printers	■ ■ ■	
LCN Coaxial Cables (Trunks)	■ ■ ■	
6100 Line Interface Units (LIUs)	■ ■ ■	
		CABINETS
		MAX DISTANCE BETWEEN COMPUTERS
		(A) 1 - 500 FT
		(B) 501 - 1000 FT
		(C) 1001 - 1500 FT
		(D) 1501 - 2500 FT
		(E) 2501 - 4000 FT
		(F) 4001 - 5000 FT

SCREEN 5

SYSTEM COMPONENTS	HARDWARE	LCN INTERFACE ADAPTERS
Processors	■ ■ ■	PERKIN-ELMER
Centronics Printers	■ ■ ■	BURROUGHS 4800
Tandem CRTs	■ ■ ■	BURROUGHS 4900
128 MB Disks	■ ■ ■	IBM System
240 MB Disks	■ ■ ■	UNIVAC System
540 MB Disks	■ ■ ■	FIPS Standard
Non-6100 ASYNC Controllers	■ ■ ■	Tandem HYPERchannel
Non-6100 ASYNC Extension Boards	■ ■ ■	
Bit SYNC Lines	■ ■ ■	
Byte SYNC Lines	■ ■ ■	
Tri-Density Tape Drives	■ ■ ■	
Reader/Punches	■ ■ ■	
Card Readers	■ ■ ■	
1000 LPM Printers	■ ■ ■	
600 LPM Printers	■ ■ ■	
LCN Coaxial Cables (Trunks)	■ ■ ■	
6100 Line Interface Units (LIUs)	■ ■ ■	
		CABINETS
		Patch Panel
		System
		Expansion
		MAX DISTANCE BETWEEN COMPUTERS
		(A) 1 - 500 FT
		(B) 501 - 1000 FT
		(C) 1001 - 1500 FT
		(D) 1501 - 2500 FT

SCREEN 6

SCREEN 7

SCREEN 8

1 - 2 - 3

Copyright (C) 1982,1983
Lotus Development Corporation
All Rights Reserved
Release 1A
★

(Press Any Key To Continue)

SCREEN 9

A1:

READY

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							

SCREEN 10

APPENDIX A: USER'S MANUAL

Page A2-55

A1:
Enter name of file to retrieve:
SKELETON MAINTORD

MENU

A B C D E F G

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

SCREEN 11

A1:

READY

A B C D E

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

SCREEN 12

A1: '

READY

	A	B	C	D	E
1					
2					
3					
4	Site:	44	NSC SAN DIEGO, CA		
5					
6					
7	Hardware				
8					
9					
10					
11	Contract Feature				
12	Line No. Numbers		Description	Qty	Unit Price
13					
14	440101	010201	NS-TXP, 2 MEG	8	86760.00
15	440102	010301	2 MEG MEMORY	8	19800.00
16	440104	013001	OSP WITH 6530	1	13387.50
17	440105	013101	CENTRONIX PRINTER	2	1615.50
18	440106	013201	6530 CRT	17	2317.50
19	440107	013202	PRINTER INTERFACE	1	409.50
20	440108	015001	PATCH PANEL CABINET	2	2250.00

SCREEN 13

PROCESS SELECTION MENU

Stock Point Logistics Integrated Communications Environment

SPLICE

- 1 - Load a new DELIVERY ORDER into the database
- 2 - Perform maintenance on the EQUIPMENT File
- 3 - Perform maintenance on the EQUIPMENT DESCRIPTION File
- 4 - Perform maintenance on the SITE NAME File
- 5 - Perform maintenance on the MANUAL File
- 6 - Perform maintenance on the SERIAL NUMBER File
- 7 - Generate REPORTS for the Project, a Site or Equipment
- 8 - Generate a MAINTENANCE DELIVERY ORDER for a SPLICE Site
- 9 - Generate MAILING LABELS for all SPLICE Sites

- 0 - RETURN to the Function Selection Menu

Please enter your choice: ■

SCREEN 14

DELIVERY ORDER LOAD MENU	
[1]	Load a new delivery order
[2]	Return to the Main Menu

SCREEN 15

DELIVERY ORDER LOADING SELECTION MENU	
LOTUS 1-2-3 output file name to load:	■■■■■■■■.PRN
Effective Date of the Delivery Order:	■■■■■ YYMMDD
Site Number on the Delivery Order:	■■
Enter the Site Number to be loaded:	■■
Do you want to enter another Delivery Order? (Yes or No): ■	

SCREEN 16

Current Record #: ■■■■

SCREEN 21

1. To edit the NOTES field, ensure the cursor is on the word "memo" and press the <CTRL> and "PgDn" keys together.
2. To EXIT the internal editor and SAVE the changes made to the NOTES field, press the <CTRL> and "W" keys together.
3. To EXIT the internal editor WITHOUT SAVING the changes made to the NOTES field, press the <ESC> key. This will return you to the full screen mode for the record being changed.
4. To SAVE the changes made by the internal editor and return to the configuration program, press the <CTRL> and "W" keys together.
5. To return to the configuration program WITHOUT SAVING the changes made by the internal editor, press the <ESC> key.

SCREEN 22

Current Record #: ■■■■■

SCREEN 23

```
[1]  Modify DataBase Entries
[2]  Review Existing Records
[3]  Return to the Main Selection Menu
```

SCREEN 24

SCREEN 27

SCREEN 28



Current Record #: ■■■■■

```
Contract Line Item Number.(CLIN):      #####  
Feature Number:                        #####  
CLIN Nomenclature/Description:         #####  
Quantity Ordered:                      ###
```

Serial Number: ■■■■■■■■

SCREEN 33

If all three of these elements are not known, the user should terminate the update process and request a DATE LEVEL REPORT for the site number in question (Option "7" on the PROCESS SELECTION MENU followed by option "3" on the REPORT BY TYPE MENU). Any one of the three options will enable the user to view all three of the data elements needed for the Serial Number Update process. Once all three data elements have been obtained, the user can then select the Serial Number Update option.

- [1] Continue with the Serial Number Update process.
- [2] Exit the Serial Number Update process to obtain a DATE LEVEL Report and the three required data elements.

119

Current Record #: ■■■■■

```
Contract Line Item Number (CLIN):      [REDACTED]
Feature Number:                         [REDACTED]
CLIN Nomenclature/Description:         [REDACTED]
Quantity Ordered:                      [REDACTED]
```

Serial Number sub-record 000 of 000

Serial Number:

Enter N - next record, P - previous record or X - exit: ■

SCREEN 35

```
[1] Overall PROJECT Report
[2] Report for a particular SITE
[3] Report for a DELIVERY ORDER issued
    on a particular date
[4] Return to the Report Level Menu
```

SCREEN 36

ATTENTION!

1. Turn on your printer.
2. Insert paper.
3. Position to top edge.

Press ENTER to continue

SCREEN 37

PROJECT LEVEL REPORTS

- [1] Report by EQUIPMENT Type
- [2] Report by SERIAL NUMBER
- [3] Return to the Reports Level Menu

SCREEN 38

EQUIPMENT PROJECT LEVEL REPORT

Do you want a printed report? (Yes or No): N

Enter C to continue or X to exit:

SCREEN 39

EQUIPMENT PROJECT LEVEL REPORT

CLIN	FEATURE#	DESCRIPTION	QTY
0001	000101	SITE POWER PREPARATIONS	2
0101	010201	NS-TXP, 2 MEG	22
0102	010301	2 MEG MEMORY	22
0104	013001	OSP WITH 6530	2
0105	013101	CENTRONIX PRINTER	5
0106	013201	6530 CRT	50
0107	013202	PRINTER INTERFACE	2
0108	015001	PATCH PANEL CABINET	5
0109	015101	SYSTEMS CABINET	3
0110	015201	I/O POWER MODULE	4
0109	015301	EXPANSION CABINET	2
0112	016001	DISC PATCH PANEL	1
0113	016101	THL PATCH PANEL	1
0114	016201	ASYNCH PATCH PANEL	1
0115	016301	SYNCH PATCHPANEL	4

Enter C to continue or X to exit: C

SCREEN 40

EQUIPMENT SERIAL NUMBER PROJECT LEVEL REPORT

Do you want a printed report? (Yes or No): N

Enter C to continue or X to exit:

SCREEN 41

EQUIPMENT SERIAL NUMBER PROJECT LEVEL REPORT

SITE	CLIN	FEATURE#	DESCRIPTION	EFFECT DATE	TOT QTY	COMPT QTY	SERIAL NUMBER
01	0102	010301	2 MEG MEMORY	841127	9	4	
01	0102	010301	2 MEG MEMORY	841127	9	3	
01	0102	010301	2 MEG MEMORY	841127	9	2	
01	0102	010301	2 MEG MEMORY	841127	9	1	
01	0104	013001	OSP WITH 6530	841127	1	1	
01	0105	013101	CENTRONIX PRINTER	841127	2	2	
01	0105	013101	CENTRONIX PRINTER	841127	2	1	
01	0106	013201	6530 CRT	841127	25	25	
01	0106	013201	6530 CRT	841127	25	24	
01	0106	013201	6530 CRT	841127	25	23	
01	0106	013201	6530 CRT	841127	25	22	
01	0106	013201	6530 CRT	841127	25	21	
01	0106	013201	6530 CRT	841127	25	20	
01	0106	013201	6530 CRT	841127	25	19	
01	0106	013201	6530 CRT	841127	25	18	

Enter C to continue or X to exit: C

SCREEN 42

SITE LEVEL REPORTS	
[1]	Report by EQUIPMENT type
[2]	Report of MANUALS
[3]	Report by SERIAL NUMBERS
[4]	Return to the Reports Level Menu

SCREEN 43

EQUIPMENT SITE LEVEL REPORT
<p>Enter site number for which the report is desired: 01</p>
<p>Enter C to continue or X to exit:</p>

SCREEN 44

EQUIPMENT SITE LEVEL REPORT

Do you want a printed report? (Yes or No): N

Enter C to continue or X to exit:

SCREEN 45

EQUIPMENT SITE LEVEL REPORT

SITE	CLIN	FEATURE#	DESCRIPTION	QTY
40	0001	000101	SITE POWER PREPARATIONS	1
40	0101	010201	NS-TXP, 2 MEG	13
40	0102	010301	2 MEG MEMORY	13
40	0104	013001	OSP WITH 6530	1
40	0105	013101	CENTRONIX PRINTER	3
40	0106	013201	6530 CRT	25
40	0107	013202	PRINTER INTERFACE	1
40	0108	015001	PATCH PANEL CABINET	3
40	0109	015101	SYSTEMS CABINET	4
40	0110	015201	I/O POWER MODULE	12
40	0109	015301	EXPANSION CABINET	1
40	0112	016001	DISC PATCH PANEL	5
40	0113	016101	THL PATCH PANEL	1
40	0114	016201	ASYNCH PATCH PANEL	2
40	0115	016301	SYNCH PATCHPANEL	2

Enter C to continue or X to exit: C

SCREEN 46

SITE LEVEL MANUAL REPORT
Enter site number for which the report is desired: 01
Enter C to continue or X to exit:

SCREEN 47

SITE LEVEL MANUAL REPORT
Do you want a printed report? (Yes or No): N
Enter C to continue or X to exit:

SCREEN 48

SITE LEVEL MANUAL REPORT				
SITE	CLIN	FEATURE#	DESCRIPTION	MANUAL DESCRIPTION
40	0101	010201	NS-TXP, 2 MEG	
40	0102	010301	2 MEG MEMORY	
40	0104	013001	OSP WITH 6530	
40	0105	013101	CENTRONIX PRINTER	
40	0106	013201	6530 CRT	
40	0107	013202	PRINTER INTERFACE	
40	0108	015001	PATCH PANEL CABINET	
40	0109	015101	SYSTEMS CABINET	
40	0110	015201	I/O POWER MODULE	
40	0109	015301	EXPANSION CABINET	
40	0112	016001	DISC PATCH PANEL	
40	0113	016101	THL PATCH PANEL	
40	0114	016201	ASYNCH PATCH PANEL	
40	0115	016301	SYNCH PATCHPANEL	
40	1101	110101	DISC CONTROLLER	
Enter C to continue or X to exit: C				

SCREEN 49

SITE SERIAL NUMBER REPORT	
Enter site number for which the report is desired: 01	
Enter C to continue or X to exit:	

SCREEN 50

SITE SERIAL NUMBER REPORT

Do you want a printed report? (Yes or No): N

Enter C to continue or X to exit:

SCREEN 51

SITE SERIAL NUMBER REPORT

SITE	CLIN	FEATURE#	DESCRIPTION	EFFECT DATE	TOT QTY	COMPT QTY	SERIAL NUMBER
40	0109	015301	EXPANSION CABINET	851207	1	1	
40	0112	016001	DISC PATCH PANEL	851207	5	5	
40	0112	016001	DISC PATCH PANEL	851207	5	4	
40	0112	016001	DISC PATCH PANEL	851207	5	3	
40	0112	016001	DISC PATCH PANEL	851207	5	2	
40	0112	016001	DISC PATCH PANEL	851207	5	1	
40	0113	016101	THL PATCH PANEL	851207	1	1	
40	0114	016201	ASYNCH PATCH PANEL	851207	2	2	
40	0114	016201	ASYNCH PATCH PANEL	851207	2	1	
40	0115	016301	SYNCH PATCHPANEL	851207	2	2	
40	0115	016301	SYNCH PATCHPANEL	851207	2	1	
40	1101	110101	DISC CONTROLLER	851207	18	18	
40	1101	110101	DISC CONTROLLER	851207	18	17	
40	1101	110101	DISC CONTROLLER	851207	18	16	
40	1101	110101	DISC CONTROLLER	851207	18	15	

Enter C to continue or X to exit: C

SCREEN 52

DELIVERY ORDER DATE LEVEL REPORT

- [1] EQUIPMENT with unit costs
- [2] EQUIPMENT without unit costs
- [3] SERIAL NUMBERS
- [4] RETURN to the Reports Level Menu

SCREEN 53

DELIVERY ORDER LEVEL REPORT

Enter site number for which the report is desired: 01

Enter C to continue or X to exit:

SCREEN 54

DELIVERY ORDER LEVEL REPORT		
EFFECTIVE DATE: 851207		
The following Delivery Order Effective Dates exist for Site 40		
851207	841127	850404
851020	851110	
Enter C to continue or X to exit:		

SCREEN 55

DELIVERY ORDER LEVEL REPORT	
EFFECTIVE DATE: 851207	
Do you want a printed report? (Yes or No): N	
Enter C to continue or X to exit:	

SCREEN 56

DELIVERY ORDER LEVEL REPORT					
EFFECTIVE DATE: 851207					
SITE	CLIN	FEATURE#	DESCRIPTION	QTY	UNIT PRICE
40	0001	000101	SITE POWER PREPARATIONS	1	101886.00
40	0101	010201	NS-TXP, 2 MEG	13	86760.00
40	0102	010301	2 MEG MEMORY	13	19800.00
40	0104	013001	OSP WITH 6530	1	13387.50
40	0105	013101	CENTRONIX PRINTER	3	1615.50
40	0106	013201	6530 CRT	25	2317.50
40	0107	013202	PRINTER INTERFACE	1	409.50
40	0108	015001	PATCH PANEL CABINET	3	2250.00
40	0109	015101	SYSTEMS CABINET	4	14220.00
40	0110	015201	I/O POWER MODULE	12	3150.00
40	0109	015301	EXPANSION CABINET	1	2250.00
40	0112	016001	DISC PATCH PANEL	5	697.50
40	0113	016101	THL PATCH PANEL	1	315.00
40	0114	016201	ASYNCH PATCH PANEL	2	697.50
40	0115	016301	SYNCH PATCHPANEL	2	697.50
Enter C to continue or X to exit: C					

SCREEN 57

DELIVERY ORDER LEVEL REPORT
Enter site number for which the report is desired: 41
Enter C to continue or X to exit:

SCREEN 58

DELIVERY ORDER LEVEL REPORT
EFFECTIVE DATE: 851207

The following Delivery Order Effective Dates exist for Site 40

851207	841127	850404
851020	851110	

Enter C to continue or X to exit:

SCREEN 59

DELIVERY ORDER LEVEL REPORT
EFFECTIVE DATE: 851207

Do you want a printed report? (Yes or No): N

Enter C to continue or X to exit:

SCREEN 60

DELIVERY ORDER LEVEL REPORT					
EFFECTIVE DATE: 851207					
SITE	CLIN	FEATURE#	DESCRIPTION	QTY	MODEL NUMBER
40	0001	000101	SITE POWER PREPARATIONS	1	
40	0101	010201	NS-TXP, 2 MEG	13	1432
40	0102	010301	2 MEG MEMORY	13	2432
40	0104	013001	OSP WITH 6530	1	3910
40	0105	013101	CENTRONIX PRINTER	3	
40	0106	013201	6530 CRT	25	6530
40	0107	013202	PRINTER INTERFACE	1	
40	0108	015001	PATCH PANEL CABINET	3	7105
40	0109	015101	SYSTEMS CABINET	4	7104
40	0110	015201	I/O POWER MODULE	12	7301
40	0109	015301	EXPANSION CABINET	1	7107
40	0112	016001	DISC PATCH PANEL	5	7504
40	0113	016101	THL PATCH PANEL	1	7506
40	0114	016201	ASYNCH PATCH PANEL	2	7501
40	0115	016301	SYNCH PATCHPANEL	2	7502
Enter C to continue or X to exit: C					

SCREEN 61

SITE SERIAL NUMBER REPORT	
Enter site number for which the report is desired: 01	
Enter C to continue or X to exit:	

SCREEN 62

SITE SERIAL NUMBER REPORT EFFECTIVE DATE: 851207		
The following Delivery Order Effective Dates exist for Site 40		
851207 851020	841127 851110	850404
Enter C to continue or X to exit:		

SCREEN 63

SITE SERIAL NUMBER REPORT EFFECTIVE DATE: 851207
Do you want a printed report? (Yes or No): H
Enter C to continue or X to exit:

SCREEN 64

SITE SERIAL NUMBER REPORT						
EFFECTIVE DATE: 851207						
SITE	CLIN	FEATURE#	DESCRIPTION	EFFECT DATE	TOT COMPT QTY QTY	SERIAL NUMBER
40	0001	000101	SITE POWER PREPARATIONS	851207	1	1
40	0101	010201	NS-TXP, 2 MEG	851207	13	13
40	0101	010201	NS-TXP, 2 MEG	851207	13	12
40	0101	010201	NS-TXP, 2 MEG	851207	13	11
40	0101	010201	NS-TXP, 2 MEG	851207	13	10
40	0101	010201	NS-TXP, 2 MEG	851207	13	9
40	0101	010201	NS-TXP, 2 MEG	851207	13	8
40	0101	010201	NS-TXP, 2 MEG	851207	13	7
40	0101	010201	NS-TXP, 2 MEG	851207	13	6
40	0101	010201	NS-TXP, 2 MEG	851207	13	5
40	0101	010201	NS-TXP, 2 MEG	851207	13	4
40	0101	010201	NS-TXP, 2 MEG	851207	13	3
40	0101	010201	NS-TXP, 2 MEG	851207	13	2
40	0101	010201	NS-TXP, 2 MEG	851207	13	1
40	0102	010301	2 MEG MEMORY	851207	13	13

Enter C to continue or X to exit: C

SCREEN 65

MAINTENANCE DELIVERY ORDER GENERATION PROGRAM

Generate maintenance Delivery Order for Site Number: 40

DISCOUNT and ESCALATION RATES

Values input are added to one (1) to generate the correct discount or escalation rates

LCN Hardware Maintenance Escalation Rate: 0.100
 LCN Software Maintenance Escalation Rate: 0.100
 SPLICENet Maintenance Discount Rate: 0.000
 Site Maintenance Escalation Rate: 0.822

File name to be imported into LOTUS 1-2-3: NEWDO.PRN

SCREEN 66

MAILING LABEL GENERATION PROGRAM

Number of copies for each label: 01

Site Number: 01

SPLICE SITE MAILING LABELS

SCREEN 67

INSTALLATION PROCEDURES

Before using the integrated Configuration Management System, make a backup copy of all five disks. Work with the backup copy and store the original disks safely away. This will allow the initial files to be restored if files are erased, damaged or an accident occurs.

Before the integrated Configuration Management System may be used, it must be installed on a micro-computer. Installation is easily accomplished using a DOS command batch file supplied on the **Initial Startup Disk**. Programs were developed on an IBM-PC environment and tested on an IBM-PC AT. System performance on other than a 100% IBM compatible configuration is unknown and without guarantee.

The integrated Configuration Management System must be run on a hard disk system configuration. This is a limitation caused by the size and number of dBASE files which are part of the system.

The integrated Configuration Management System consists of five disks, one installation disk and four system disks. Each disk is labelled to reflect the portion of the system residing on each disk. The label identifies the directory where the system files must be loaded. Three directories will be created during the installation process if they do not already exist. These directory names may not be modified. System execution is dependent on files existing in predefined locations.

Ensure the system default drive is the hard disk where the integrated system is to be loaded (ex: C> or D>, etc.). Start system installation by placing the disk labelled **Initial Startup Disk** in drive A. Type the command **STARTUP** and follow the instructions and messages displayed on the screen.

APPENDIX B

THE NAVAL SUPPLY SYSTEMS COMMAND
STOCK POINT LOGISTICS INTEGRATED COMMUNICATIONS ENVIRONMENT
(SPLICE)
SYSTEM CONFIGURER AND CONFIGURATION MANAGEMENT SYSTEM
MAINTENANCE MANUAL

Document No. BBC - 02

1 January 1986

Record of Changes

Original

1 February 1986

List of Effective Pages

Page 1 through 285

Original

Table of Contents

Record of Changes -----	2
List of Effective Pages -----	3
Table of Contents -----	4
Overall System Data Flow Diagram -----	7
Configurer Data Flow Diagram (Level 1) -----	8
File Description Example -----	9
Data Flow Description Example -----	11
Process Description Example -----	12
Configurer Structure Chart (Overview - Level 0) -----	14
Configurer Structure Chart (Level 1) -----	15
Configurer Structure Chart (Level 2) -----	16
Module Description Example -----	17
Pascal Configurer Record Descriptions -----	20
Configuration Input Data File (CONFIG.SIT) -----	22
Input Data File (COSTS.IN) -----	24
Configuration Management System Bachmann Diagram -----	28
dBASE III Data Base Structure Descriptions -----	29
dBASE III Configuration Management System Indices Composition -----	32
dBASE III Configuration Management System Program Invocation Sequences -----	33
Pascal SPLICE Configurer Program Source Code -----	34
dBASE III Configuration Management System Program Module Source Code:	

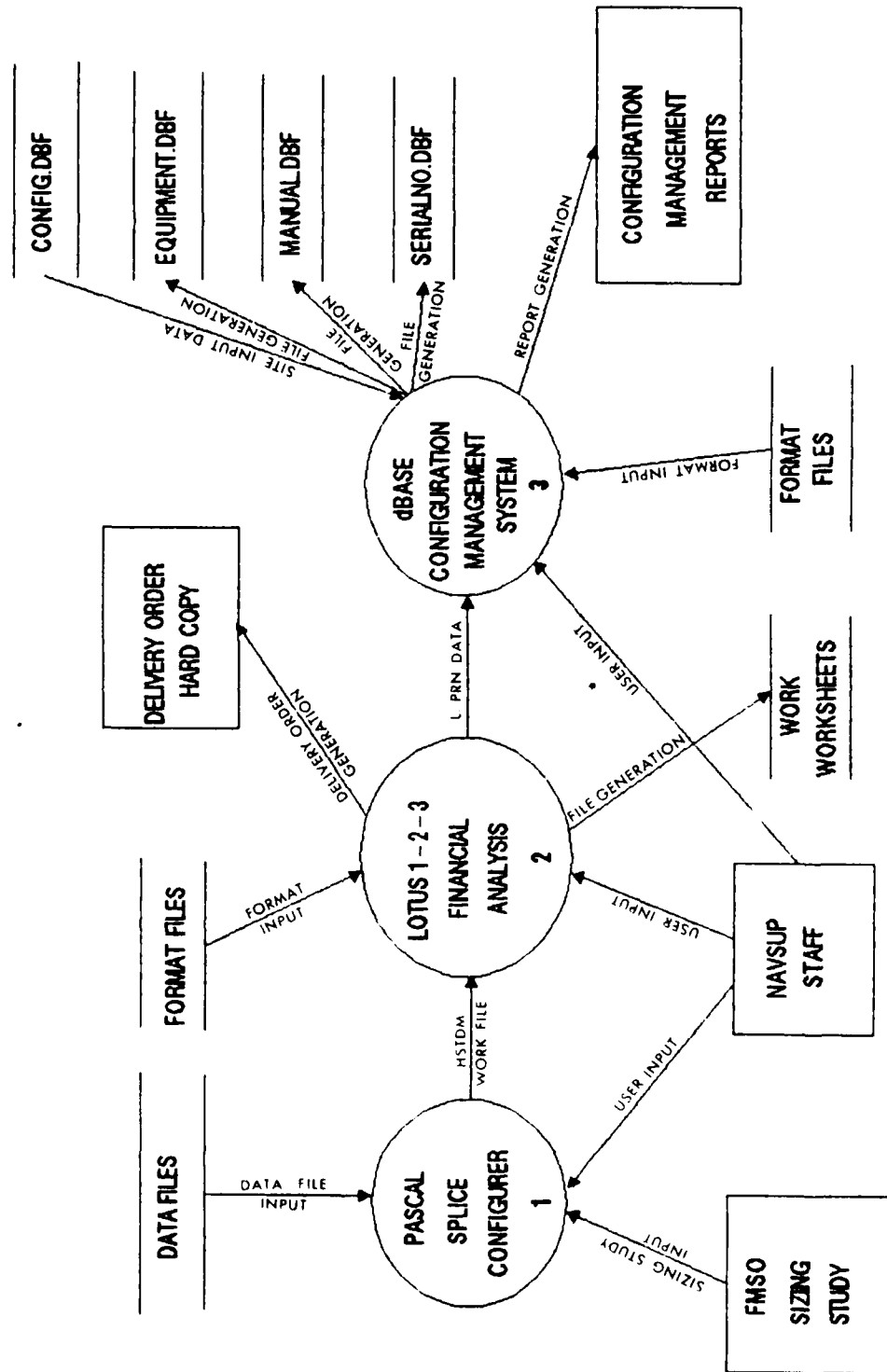
APPENDIX B: MAINTENANCE MANUAL Page 5

CONFMOD.PRG	-----	102
CONFREV.PRG	-----	104
CONFUPD.PRG	-----	108
DATERPTS.PRG	-----	114
DELAY.PRG	-----	116
DESPMOD.PRG	-----	117
DESPPREV.PRG	-----	119
DESPPUPD.PRG	-----	123
EQPD TNPC.PRG	-----	130
EQPDTPRC.PRG	-----	140
EQPPJRPT.PRG	-----	150
EQPSTRPT.PRG	-----	155
EQUIPCMD.PRG	-----	161
EQUIPREV.PRG	-----	163
EQUIPUPD.PRG	-----	169
MAINMENU.PRG	-----	176
MAINTDO.PRG	-----	179
MANULADD.PRG	-----	184
MANULCMD.PRG	-----	191
MANULDEL.PRG	-----	193
MANULREV.PRG	-----	198
MANULUPD.PRG	-----	204
MKLABELS.PRG	-----	211
MNLSTRPT.PRG	-----	216

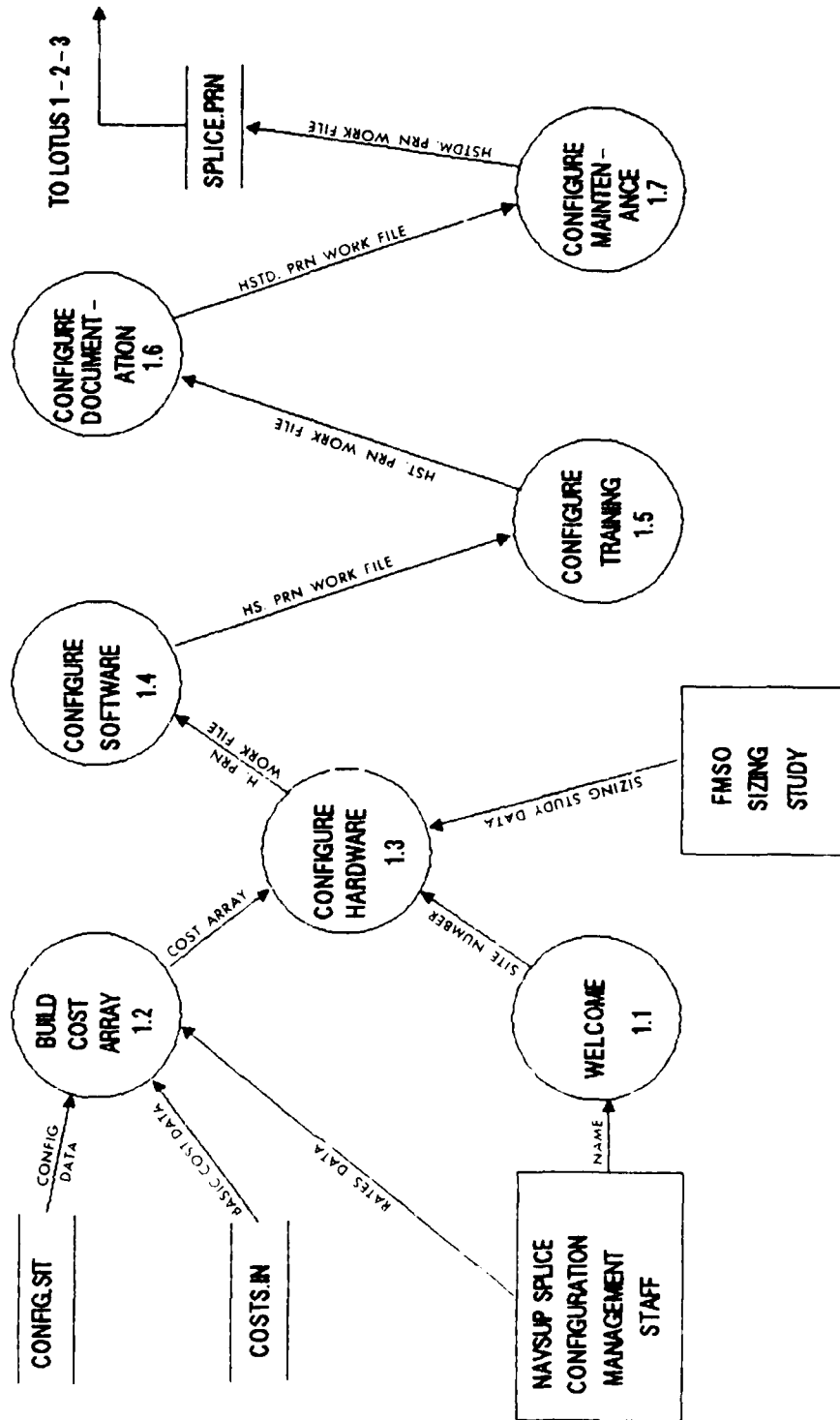
APPENDIX B: MAINTENANCE MANUAL Page 6

NEWDOADD.PRG	-----	221
NEWDCMD.PRG	-----	224
NEWDOCVT.PRG	-----	226
PROJRPTS.PRG	-----	235
REPORCMD.PRG	-----	237
SELECTOR.PRG	-----	239
SERNCBLD.PRG	-----	242
SERNOCMD.PRG	-----	244
SERNOREV.PRG	-----	246
SERNOUPD.PRG	-----	252
SITERPTS.PRG	-----	263
SNODTRPT.PRG	-----	265
SNOPJRPT.PRG	-----	275
SNOSTRPT.PRG	-----	280

OVERALL SYSTEM DATA FLOW DIAGRAM



CONFIGURER DATA FLOW DIAGRAM LEVEL 1



FILE DESCRIPTION

PROJECT: SPLICE CONFIGURER

DATE: 3 September 1985

FILE OR DATABASE NAME: CONFIG.SIT

ALIASES: None

COMPOSITION: The CONFIG.SIT file contains the site specific data associated with all the designated Stock Point Logistics Integrated Communications Environment (SPLICE) sites.

ORGANIZATION: Sequential. The structure of the file is as follows:

	<u>DATA ELEMENT</u>	<u>TYPE VARIABLE</u>
	Site Number	Integer
	Site Name	String
	Documentation Site Group	Integer
	Training Site Group	Integer
*	Maintenance Option	String
*	Maintenance Responsibility	String
	Site Type	String
	Installation Cost	Real

* **NOTES:** These data elements are not currently designated for implementation, but are specified for use in later revisions.

1. Site Number range can be from one (1) to sixty-two (62). Current only fifty-six (56) sites are designated SPLICE sites and is the upper range limit.

2. Documentation Site Group is used to restrict the maximum number of documentation sets that each site is allowed to receive.

3. Training Site Group is used to restrict the maximum number of training courses that each site is allowed to receive.

FILE DESCRIPTION (Continued)

4. Maintenance Option and Maintenance Responsibility are used together to establish the maintenance repair and response times desired by each site.

5. Site Type restricts various hardware options to certain designated sites. The value is either "S" or "M". "S" designates a site as a Stock Point which can receive all hardware/software options. "M" designates a site as a Multiple Activity Processing System (MAPS) site which is not permitted to receive Local Computer Network (e.g. HYPERchannel) components.

6. Installation Cost is a one time cost that is paid to the vendor for his initial site survey and installation preparations.

DATA FLOW DESCRIPTION

PROJECT: SPLICE CONFIGURER

DATE: 3 September 1985

DATAFLOW NAME: Config_Data

ALIASES: None

COMPOSITION: The data represented in this flow is the data coming from the input file "CONFIG.SIT". The site number selected for configuration is located within the CONFIG.SIT file and site unique information is extracted. This unique site configuration data is then used to create the site information record. This record is used to determine the maximum limits applicable to sites under configuration, as specified in the notes to the CONFIG.SIT file description. The site information record also is used to determine which repair and maintenance options are to be selected and serves to restrict certain types of options from being selected, depending upon the sites type designation. The Site Preparations Charge is taken from the CONFIG.SIT file and input as the first entry in the COSTTABLE array.

NOTES: The user was previously prompted for the number of the site to be configured.

PROCESS DESCRIPTION

PROJECT: SPLICE CONFIGURER

DATE: 3 September 1985

PROCESS NAME: Build Cost Array

PROCESS NUMBER: 1.2

PROCESS DESCRIPTION:

1. Take a feature number for each element resident in the input cost file and place it in a feature number field in the cost array.

2. Take a contract line item number for each element resident in the input cost file and place it in a contract line item number field in the cost array.

3. Take a nomenclature description for each element resident in the input cost file and place it in an item description field in the cost array.

4. Take the unit maintenance costs from the input cost file and place it in the fourth element of the cost array.

5. Take the unit purchase price from the input cost file and apply a discount rate specified by the user. Place the result in the fifth element of the cost array.

6. Take the unit installation cost from the input cost file and apply an escalation rate specified by the user. Place the result in the sixth element of the cost array.

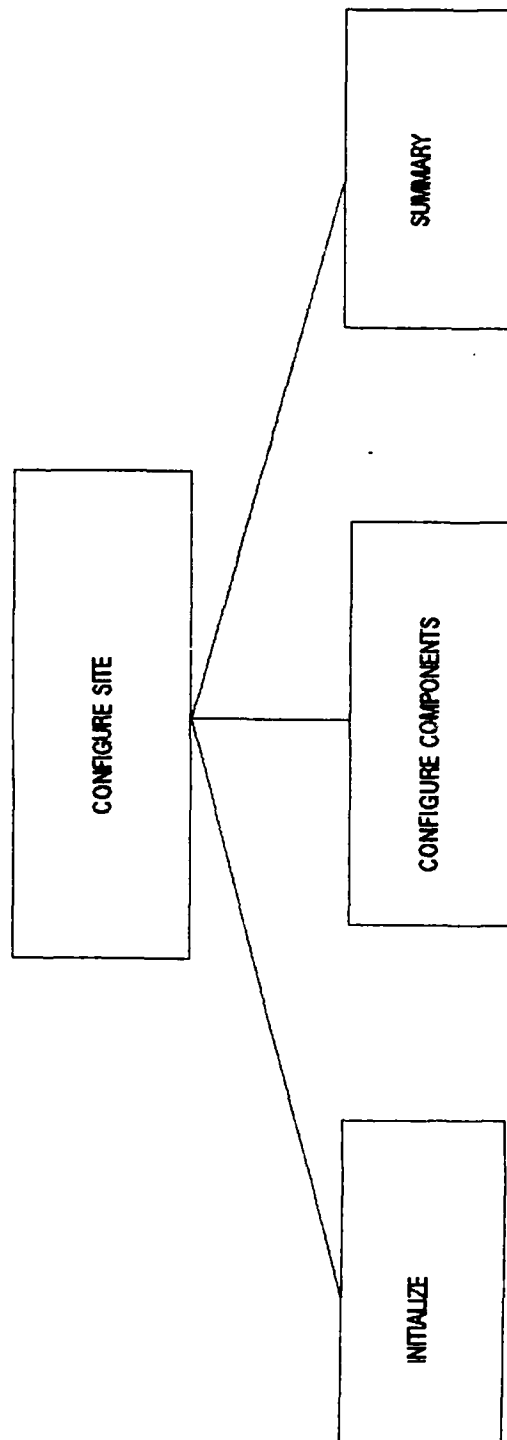
7. Take the basic monthly maintenance rate from the input cost file and apply an escalation rate specified by the user. Place the result in the seventh element of the cost array.

PROCESS DESCRIPTION (Continued)

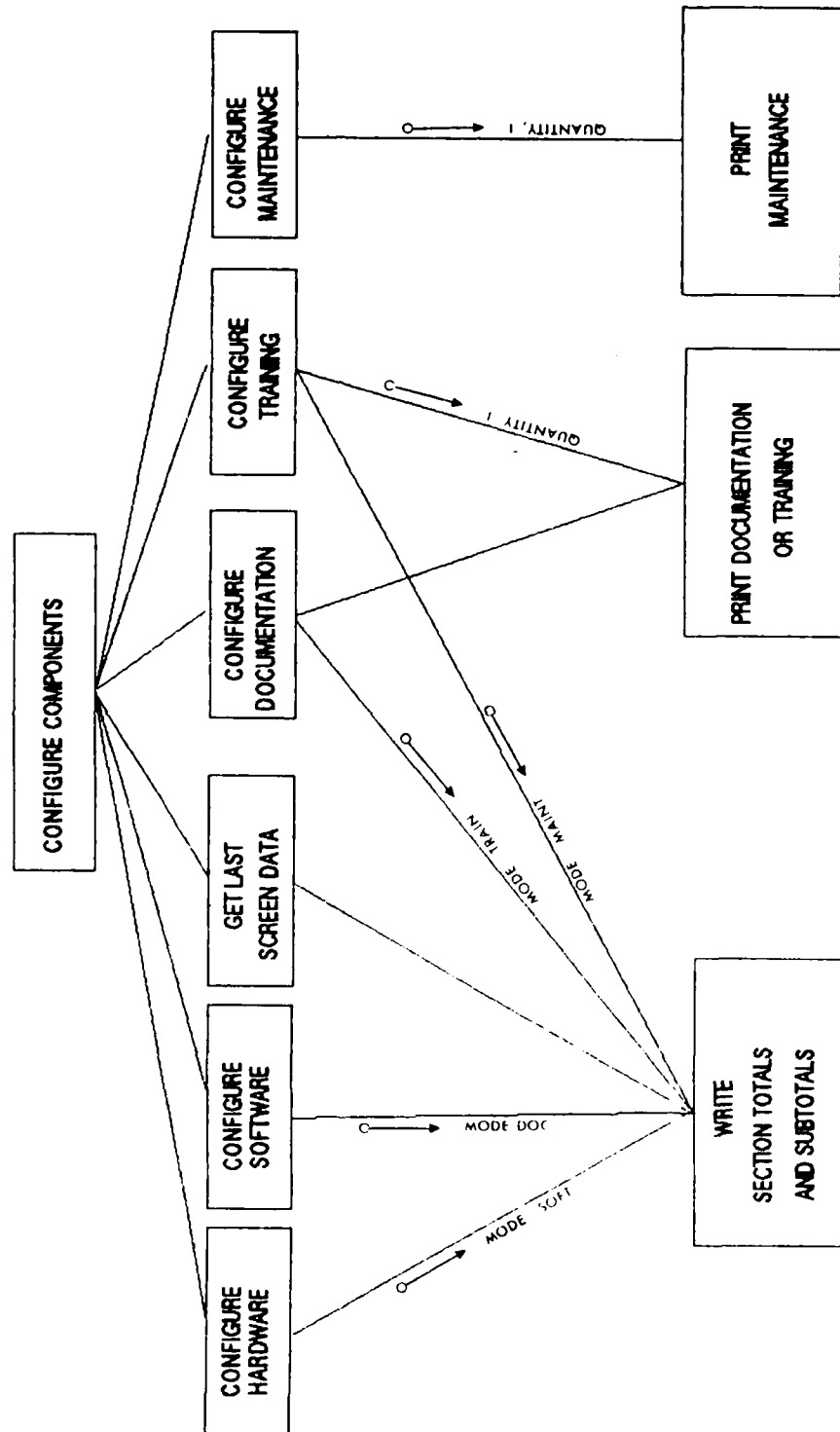
NOTES: The cost array mentioned on the previous page is a two dimensional memory array. The array contains an entry for every line item identified on the Automated Data Processing Selection Office (ADPSO) SPLICE contract. The maximum number of entries expected is two hundred. This estimate is based upon the maximum number of possible line items that may exist for available selections. The site cost array structure is planned as follows:

<u>FEATURE NUMBER</u>	<u>VARIABLE TYPE</u>
Contract Line Item Number (CLIN)	String
CLIN Description	String
Monthly Maintenance	Real
CLIN Unit Price	Real
Installation Cost	Real
Basic Monthly Maintenance Cost	Real

CONFIGURER STRUCTURE CHART OVERVIEW - LEVEL 0



CONFIGURER STRUCTURE CHART - LEVEL 1



MODULE DESCRIPTION

SPLICE DESIGN

DATE: 3 September 1985

MODULE NAME: Print_Software

MODULE PURPOSE: This routine writes software related contract line item numbers (CLINs) to the output disk file. It invokes global procedure LINE_SETUP to generate the CLIN and accumulate section and appropriation totals. The global variable Quantity is used to compute the CLIN extended price and installation costs. The procedure is not used in maintenance computations.

INPUT: None.

OUTPUT: CLIN related data elements written to output disk file are:

Line_Number	String- 7
CostTable[I].featureno	String- 8
CostTable[I].descript	String-28
Quantity	Integer-3
CostTable[I].purchprice	Real-13, 2 decimals
Extended_Price	Real-12, 2 decimals
CostTable[I].basemaint	Real- 9, 2 decimals
Maint_Factor	Real- 8, 3 decimals
Maint_Months	Integer-5
CostTable[I].basemaint * Maint_Factor * Maint_Months	Real- 9, 2 decimals
CostTable[I].instcost	Real- 8, 2 decimals
CostTable[I].instcost * Quantity	Real- 9, 2 decimals
Downtime_Credit	Real- 9, 2 decimals
CostTable[I].basemaint * Quantity * Maint_Factor	Real- 9, 2 decimals

MODULE DESCRIPTION (Continued)

PROCEDURAL DESCRIPTION:

```

Begin  [Print Software]
  CASE Type_Software of
    1: Begin  { Per Processor Software }
        Maint_Factor = Momaint_Esc_Cost
        Extended_Price = Quantity * CostTable
      End
    2: Begin  { Per Site Software }
        Maint_Factor = Momaint_Esc_Cost
        Extended_Price = CostTable[I].purchprice
      End
    3: Begin  { NETEX Software }
        Maint_Factor = 1
        Extended_Price = CostTable[I].purchprice
                          * Quantity
      End
  END [End of CASE Statement]
  CALL LINE_SETUP
  Compute_System_Downtime_Component      * See Notes
  • Compute_Downtime_Credit              * See Notes
  Write_CLIN_Data_Elements_to_Output_Disk_File
End    [Print_Software]

```

MODULE DESCRIPTION (Continued)

VARIABLES:

PROGRAM GLOBALS: See CONFIGURE_SITE module description

MODULE LOCALS: None.

PROCEDURE LOCALS:

1: Type_Software - Integer, parameter list variable, Range: 1-3, code controlling which values are assigned to the variables Maint_Factor and Extended_Price.

NOTES:

1. Computation for System_Downtime_Component:

$$\text{System_Downtime_Component} + (\text{Maint_Factor} * \text{Quantity} * \text{CostTable[I].basemaint})$$
2. Computation for Downtime_Credit:

$$(((\text{CostTable[I].purchprice} + \text{CostTable[I].instcost}) / 48) + (\text{CostTable[I].basemaint} * \text{Maint_Factor})) * 0.005$$

PASCAL CONFIGURER RECORD DESCRIPTIONS

1. COSTS.IN - file contains the individual contract line items which appear as line items on the generated delivery order.

<u>COLUMN POSITION</u>	<u>FIELD LENGTH</u>	<u>DATA ELEMENT DESCRIPTION</u>
01-04	4	Contract Line Item Number (CLIN)
05	1	Blank (Filler)
06-11	6	Contract Feature Number
12	1	Blank (Filler)
13-39	27	Component Description
40-48	9	Basic Contract Maintenance Rate
49	1	Blank (Filler)
50-60	11	Basic Contract Purchase Price
61	1	Blank (Filler)
62-69	8	Basic Contract Installation Rate
70-80	11	Blank (Filler)

NOTE: All data elements are left justified. This file is read into a memory array (COSTTABLE). The data elements are modified by the discount and escalation rates entered by the user. The file is maintained in Contract Feature Number sequence, with two exceptions. T-Text and TRANSFER line items are not in Contract Feature Number sequence. Use extreme care when adding components and corresponding line items in the source code. Line items are identified in the source code by using comments. An example of a comment is { I=6 Serial Printers }.

PASCAL CONFIGURER RECORD DESCRIPTIONS (Continued)

2. CONFIG.SIT - file contains site specific information used to determine several factors required in the configuration process.

<u>COLUMN POSITION</u>	<u>FIELD LENGTH</u>	<u>DATA ELEMENT DESCRIPTION</u>
01-02	2	Site Number
03-30	28	Site Name
31	1	Documentation Site Group
32	1	Blank (Filler)
33	1	Training Site Group
34	1	Blank (Filler)
35-38	4	Maintenance Option
39	1	Blank (Filler)
40	1	Maintenance Responsibility
41	1	Blank (Filler)
42	1	Site Type (Stock Point or MAP Site)
43	1	Blank (Filler)
44-49	6	Installation Cost
50-80	31	Blank (Filler)

NOTE: All data elements are left justified. Site specific information is read into a memory array (SITEINFO). The file is maintained in site number sequence. Site installation costs were obtained from NAVSUP SPLICE personnel. Installation costs reflect costs originally specified in the SPLICE contract. If these costs are not correct or are revised, update the site preparation charges in CONFIG.SIT prior to running the configurer.

Page 1

CONFIG.SIT Program Listing

1	01 ASO PHILADELPHIA, PA	2 2 X	A S	81735.0
2	02 FMSO MECHANICSBURG, PA	1 2 III	B S	93939.0
3	03 FMSO MECHANICSBURG, PA	4 4 I	F M	56721.0
4	04 MCAS CHERRY POINT, NC	3 3 VIII	C M	70860.0
5	05 MCAS EL TORO, CA	3 3 II	D M	76473.0
6	06 MCAF QUANTICO, VA	4 4 P	F M	59748.0
7	07 MCAS YUMA, AZ	4 4 I	F M	59748.0
8	08 NAC INDIANAPOLIS, IN	4 4 P	A M	59748.0
9	09 NARDAC JACKSONVILLE, FL	2 2 VIII	A S	188471.0
10	10 NARDAC NEW ORLEANS, LA	2 2 P	A S	73918.0
11	11 NARDAC NORFOLK, VA	2 2 VIII	A S	74913.0
12	12 NARDAC PENSACOLA, FL	2 2 VIII	A S	76523.0
13	13 NARDAC SAN DIEGO, CA	2 2 VIII	A S	74829.0
14	14 NARDAC SAN FRANCISCO, CA	2 2 VIII	A S	75967.0
15	15 NARDAC WASHINGTON, DC	4 4 P	F S	59748.0
16	16 NAS BARBERS POINT, HI	4 4 I	F M	59748.0
17	17 NAS BRUNSWICK, ME	4 4 I	F M	59748.0
18	18 NAS CECIL FIELD, FL	4 4 I	F M	59748.0
19	19 NAS KEY WEST, FL	4 4 I	F M	59748.0
20	20 NAEC LAKE HURST, NJ	4 4 I	F M	59748.0
21	21 NAS MEMPHIS, TN	4 4 I	F M	59748.0
22	22 NAS MIRAMAR, CA	4 4 I	F M	59748.0
23	23 NAS OCEANA, VA	4 4 I	F M	59748.0
24	24 NAS PENSACOLA, FL	4 4 I	F M	59748.0
25	25 NAS WHIDBEY ISLAND, WA	3 3 P	D M	68448.0
26	26 NATC PATUXENT RIVER, MD	3 3 II	D M	63841.0
27	27 PMTC POINT MUGU, CA	4 4 I	F M	59748.0
28	28 NAVDAF Corpus Christi, TX	4 4 I	F M	59748.0
29	29 NAVDAF GREAT LAKES, IL	4 4 I	F M	59748.0
30	30 NAVDAF LEMOORE, CA	4 4 I	F M	59748.0
31	31 NAVDAF MOFFETT FIELD, CA	4 4 I	F M	59748.0
32	32 NAVDAF ORLANDO, FL	4 4 I	F M	59748.0
33	33 NRCC LONG BEACH, CA	4 4 I	F S	57816.0
34	34 NRCC NEWPORT, RI	4 4 I	F S	57816.0
35	35 NRCC PHILADELPHIA, PA	4 4 I	F S	57816.0
36	36 NRCC WASHINGTON, D.C.	4 4 I	F S	57816.0
37	37 NUWES KEYPORT, WA	4 4 I	F M	59748.0
38	38 NAVSTA Mayport, FL	4 4 I	F M	59748.0
39	39 NSC CHARLESTON, SC	2 2 VIII	A S	78279.0
40	40 NSC NORFOLK, VA	2 2 X	A S	101886.0
41	41 NSC OAKLAND, CA	2 2 V	A S	94646.0
42	42 NSC PEARL HARBOR, HI	2 2 VIII	A S	89493.0
43	43 NSC PUGET SOUND, WA	2 2 VIII	A S	89493.0
44	44 NSC SAN DIEGO, CA	2 2 V	A S	95520.0
45	45 NSD GUAM	3 3 VII	E S	101824.0
46	46 NSD SUBIC BAY, PI	3 3 P	E S	102835.0
47	47 NSD YOKOSUKA, JAPAN	3 3 VII	E S	108835.0
48	48 NSY PHILADELPHIA, PA	4 4 I	E M	59748.0
49	49 NSY PORTSMOUTH, NH	4 4 I	E M	59748.0
50	50 NTC SAN DIEGO, CA	4 4 I	F M	59748.0

Page 2

CONFIG.SIT Program Listing

51	51	SPCC MECHANICSBURG, PA	2	2	X	A	S	95520.0
52	52	SUBASE KINGS BAY, GA	4	4	VIII	D	M	59748.0
53	53	SUBASE NEW LONDON, CN	4	4	I	F	M	59748.0
54	54	SUBASE PEARL HARBOR, HI	4	4	I	F	M	59748.0
55	55	SWFPAC BREMERTON, WA	3	3	VI	E	M	88507.0
56	56	TRF BANGOR, WA	4	4	P	E	S	88507.0
57	57	SWFPAC KINGS BAY, GA	3	3	VI	E	M	88507.0
58	58	TRF KINGS BAY, GA	4	4	P	E	S	88507.0

APPENDIX B: MAINTENANCE MANUAL Page 24

Page 1

COSTS.IN Program Listing

1	0001	000101	SITE POWER PREPARATIONS	0.0	0.0	0.0
2	0101	010201	NS-TXP, 2 MEG	439.2	96400.0	700.0
3	0102	010301	2 MEG MEMORY	89.06	22000.0	113.0
4	0103	012401	FLTG PT ARITH		2000.0	100.0
5	0104	013001	OSP WITH 6530	198.86	14875.0	300.0
6	0105	013101	CENTRONIX PRINTER	30.0	1795.0	0.0
7	0106	013201	6530 CRT	35.38	2575.0	100.0
8	0107	013202	PRINTER INTERFACE	2.44	455.0	0.0
9	0108	015001	PATCH PANEL CABINET	0.0	2500.0	400.0
10	0109	015101	SYSTEMS CABINET	202.52	15800.0	600.0
11	0110	015201	I/O POWER MODULE	48.8	3500.0	600.0
12	0109	015301	EXPANSION CABINET	0.0	2500.0	400.0
13	0112	016001	DISC PATCH PANEL	0.0	775.0	75.0
14	0113	016101	THL PATCH PANEL	0.0	350.0	0.0
15	0114	016201	ASYNCH PATCH PANEL	7.32	775.0	75.0
16	0115	016301	SYNC PATCHPANEL	7.32	775.0	75.0
17	1101	110101	DISC CONTROLLER	70.76	10500.0	200.0
18	1201	120201	DISC, WINCHESTER, 128MB	123.22	19500.0	325.0
19	1202	120301	DRAWER, WINCHESTER, 128MB	123.22	16500.0	325.0
20	1301	130201	DISC, MOVING HEAD, 240MB	253.76	26500.0	450.0
21	1401	140201	DISC, WINCHESTER, 540MB	395.28	39500.0	625.0
22	2101	210101	TAPE CONTROLLER	41.48	6100.0	100.0
23	2102	210201	TAPE DRIVE/FORMATTER	469.7	47500.0	475.0
24	24	240101	CARD RDR/PNCH	191.0	20442.0	75.0
25	24	240201	CARD READER	56.12	5600.0	175.0
26	2701	270101	LP/CR CONTROLLER	24.4	2800.0	188.0
27	2702	270201	1000 LPM PRINTER	202.52	20000.0	100.0
28	27	270301	600 LPM PRINTER	202.52	14000.0	100.0
29	31	310101	INTRPROC BUS(INCL.W/010101)	0.0	0.0	0.0
30	31	310201	FIBER OPTIC LINK CNTRL	610.00	35000.0	450.0
31	31	310202	FIBER OPTIC CABLES	0.0	3750.0	0.0
32	3201	320101	TANDEM/P-E HC ADAP	215.0	38940.0	0.0
33	3202	320102	HC ADAPTER 2ND TRUNK INTER	28.0	4705.0	0.0
34	3203	320201	HC CABINET (3 ADAP)	14.0	3760.0	0.0
35	3204	320301	THL CONTROLLER	194.22	14900.0	300.0
36	3207	320400	HC TRUNK, 500 FT.	0.0	400.0	0.0
37	3207	320401	HC TRUNK, 1000 FT	0.0	800.0	0.0
38	32	320402	HC TRUNK, 1500 FT.	0.0	3075.0	0.0
39	32	320403	HC TRUNK, 2500 FT.	0.0	6250.0	0.0
40	32	320404	HC TRUNK, 4000 FT.	0.0	12600.0	0.0
41	32	320405	HC TRUNK, 5000 FT.	0.0	22750.0	0.0
42	3301	330101	BURROUGHS HTC HC	215.0	38440.0	0.0
43	3302	330201	BURROUGHS DLF HC	215.0	38440.0	0.0
44	3303	330301	ECBDIC-ASCII RAM	16.0	3225.0	0.0
45	34	340301	HC PROC 1/F(P.E./INTERDATA)	19.0	4060.0	0.0
46	36	360101	HC ADAPTER (IBM 360/370)	215.0	39515.0	0.0
47	37	370101	HC ADAPTER(UNIVAC 1100,490)	215.0	38440.0	0.0
48	41	410101	HC ADAPTER(FIPS DEVICE)	215.0	39515.0	0.0
49	42	420301	HC PROC 1/F (MINI-COMPUTER)	19.0	4000.0	0.0
50	4501	450101	ASYNCH CNTR	21.96	3600.0	125.0

APPENDIX B: MAINTENANCE MANUAL Page 25

Page 2

COSTS.IN Program Listing

51	4502	450102	ASYNCH EXTENSION BOARD	26.84	4300.0	188.0
52	4503	450103	AUTOMATIC CALLING UNIT	8.19	1540.0	0.0
53	4504	450301	COMM SUBSYSTEM BASE	160.0	23673.0	630.0
54	4505	450302	BASE ADD-ON	124.0	19374.0	500.0
55	4506	450303	RS-232 LIU/CABLE	12.0	1869.0	157.0
56	4507	450304	6100 CABLE/30M	0.0	145.0	0.0
57	4508	450305	6100 CABLE/45M	0.0	160.0	0.0
58	4509	450306	6100 CABLE/60M	0.0	175.0	0.0
59	4601	460101	BIT SYNCH CNTR	50.02	6059.0	125.0
60	4602	460201	BYTE SYNCH CNTR	35.38	5800.0	100.0
61	4701	470101	COMM.PATCH PANEL/LINE MON	140.4	6653.0	100.0
62	4702	470201	ARCLI	7.02	5145.0	5.0
63	5101	510101	GUARDIAN OS	158.6	3500.0	125.0
64	5102	510201	BATCH SUBSYSTEM	61.0	4444.0	0.0
65	5103	510301	FDC SYS UTILITIES	125.0	1000.0	0.0
66	5201	520101	ENCOMPASS	323.3	8000.0	150.0
67	5202	520102	ENABLE (P/O 520101)	67.1	1500.0	0.0
68	5203	520103	ENFORM (P/O 520101)	85.4	2000.0	0.0
69	5204	520104	PATHWAY (P/O 520101)	103.7	2500.0	0.0
70	5205	520105	TMF (P/O 520101)	122.0	2500.0	0.0
71	5206	520106	DDL (P/O 520101)	36.6	500.0	0.0
72	5207	520107	FDC TPS SAS	240.0	3500.0	0.0
73	5308	530102	ENSCRIBE (P/O 510101)	0.0	0.0	0.0
74	5309	530103	SORT/MERGE (P/O 510101)	0.0	0.0	0.0
75	5310	530104	FUP(FILE UTIL PRG,P/O510101)	0.0	0.0	0.0
76	5311	530105	PUP(PERIP UTL PRG,P/O510101)	0.0	0.0	0.0
77	5312	530106	BACKUP/RESTORE(P/O 510101)	0.0	0.0	0.0
78	5313	530107	FILE SYSTEM SECURITY	600.0	5000.0	0.0
79	5314	530108	SYSTEM CARD READER SUPPORT	0.0	0.0	0.0
80	5401	540201	SPOOLER (P/O 510101)	24.4	500.0	0.0
81	5502	550102	ENVOY (P/O 510101)	0.0	0.0	0.0
82	5502	550103	CUP(COM UTL PRG,P/O 510101)	0.0	0.0	0.0
83	5503	550201	EXPAND	122.0	2000.0	50.0
84	5504	550301	EXCHANGE RJE HASP	24.4	500.0	50.0
85	5505	550401	AM3270 ACCESS METHOD	24.4	500.0	50.0
86	5506	550501	X25 ACCESS METHOD	24.4	500.0	50.0
87	5507	550601	HYPER LINK ACC MD(P/O510101)	0.0	0.0	0.0
88	5508	550602	LCN FUP SUPPORT	0.0	0.0	0.0
89	5509	550701	DELETED FDC CRT SUPPORT	350.0	13000.0	0.0
90	5510	550702	6100 ATP	27.0	430.0	50.0
91	55	550703	6100 BSC	27.0	430.0	50.0
92	55	550704	6100 ADCCP	27.0	430.0	50.0
93	55	550705	6100 TINET	27.0	430.0	50.0
94	5528	550706	BURR POLL/SELECT	27.0	430.0	50.0
95	5530	550707	SNAX AND SNAX/HLS	27.0	430.0	102.0
96	5528	550708	TR 3271	54.0	860.0	78.0
97	5530	550709	AM 6520	27.0	430.0	50.0
98	55	550710	FDC SNA INTERFACE PACKAGE	350.0	84000.0	0.0
99	55	550711	FDC DLANET INTERFACE PKG	400.0	25000.0	0.0
100	5511	550801	BURROUGHS HTC NETEX	156.4	391.0	0.0

APPENDIX B: MAINTENANCE MANUAL Page 26

Page 3

COSTS.IN Program Listing

101	5512	550802	DELETED HTC PRESENTATION	450.0	0.0	0.0
102	55	550803	CIP, BURROUGHS HTC	450.0	22500.0	0.0
103	55	550901	BURROUGHS DLP NETEX	680.0	720.0	0.0
104	55	550902	DELETED DLP PRESENTATION	450.0	0.0	0.0
105	55	550903	CIP, BURROUGHS DLP	450.0	22500.0	0.0
106	55	551001	PE 3200 NETEX	680.0	725.0	0.0
107	55	551002	DELETED PE3200 PRESENTATN	450.0	0.0	0.0
108	55	551003	CIP, PERKIN-ELMER	450.0	22500.0	0.0
109	55	551101	IBM NETEX	800.0	850.0	0.0
110	55	551102	DELETED IBM PRESENTATION	450.0	0.0	0.0
111	55	551103	CIP, IBM MVS	450.0	22500.0	0.0
112	55	551201	UNIVAC 1100 NETEX	800.0	850.0	0.0
113	55	551202	DELETED UNIVAC 1100 PRESEN	450.0	0.0	0.0
114	55	551203	CIP, UNIVAC	450.0	22500.0	0.0
115	5520	551301	TANDEM NETEX	326.4	816.0	0.0
116	5521	551302	DELETED TANDEM PRESENTATION	450.0	0.0	0.0
117	55	551303	CCP, TANDEM	550.0	27500.0	0.0
118	55	551304	CEM, TANDEM	475.0	22500.0	0.0
119	5522	551401	DELETED DDN INTERFACE	24.4	500.0	0.0
120	5523	551402	DELETED DDN SVC INTERFACE	350.0	13000.0	0.0
121	55	551403	DDN INTERFACE SUBSYSTEM	750.0	32000.0	0.0
122	55	551500	NETWORK MGMNT FACILITY GRP	324.0	13200.0	0.0
123	55	551501	NMF BASE FACILITY	180.0	6000.0	0.0
124	55	551502	NMF PERFORMANCE MONITORING	75.0	3500.0	0.0
125	55	551503	NMF DIAGNOSTIC MONITORING	75.0	3500.0	0.0
126	55	551504	NMF ACCOUNTING APPLICATION	75.0	3500.0	0.0
127	6101	610102	EDIT (P/O 510101)	0.0	0.0	0.0
128	6102	610103	TGAL (P/O 510101)	0.0	0.0	0.0
129	6103	610201	FILE COMPARSION UTILITY	0.0	0.0	0.0
130	6201	621001	COBOL	85.4	500.0	50.0
131	62	622001	TAL(P/O 510101)	0.0	0.0	0.0
132	62	623001	BLOCK STRUCTURED LANGUAGE	61.0	170.0	50.0
133	62	624001	FORTRN-ANSI 78	73.2	500.0	50.0
134	62	626001	BINDER (P/O 510101)	0.0	0.0	0.0
135	62	627001	ENFORM (P/O 510101)	85.4	2000.0	0.0
136	62	627002	DDL (P/O 510101)	36.6	500.0	0.0
137	62	628001	BINDER (P/O 510101)	0.0	0.0	0.0
138	62	629001	FUP (P/O 510101)	0.0	0.0	0.0
139	62	629002	EDIT (P/O 510101)	0.0	0.0	0.0
140	63	630101	BINDER (P/O 510101)	0.0	0.0	0.0
141	63	630102	OSP (P/O 510101)	0.0	0.0	0.0
142	63	630103	ENCORE (P/O 510101)	0.0	0.0	0.0
143	63	630104	XREF (P/O 510101)	0.0	0.0	0.0
144	63	630105	LOADFILE (P/O 510101)	0.0	0.0	0.0
145	64	640101	XRAY (P/O 510101)	0.0	0.0	0.0
146	64	640301	ENABLE (P/O 510101)	67.1	1500.0	0.0
147	65	650101	RUNTIME MON SYS (P/O 510101)	0.0	0.0	0.0
148	66	660101	TANDEM DIAG SBSYS(P/O510101)	0.0	0.0	0.0
149	8601	860101	TRANSFER	122.0	2000.0	0.0
150	8602	860201	T-TEXT	0.0	0.0	0.0

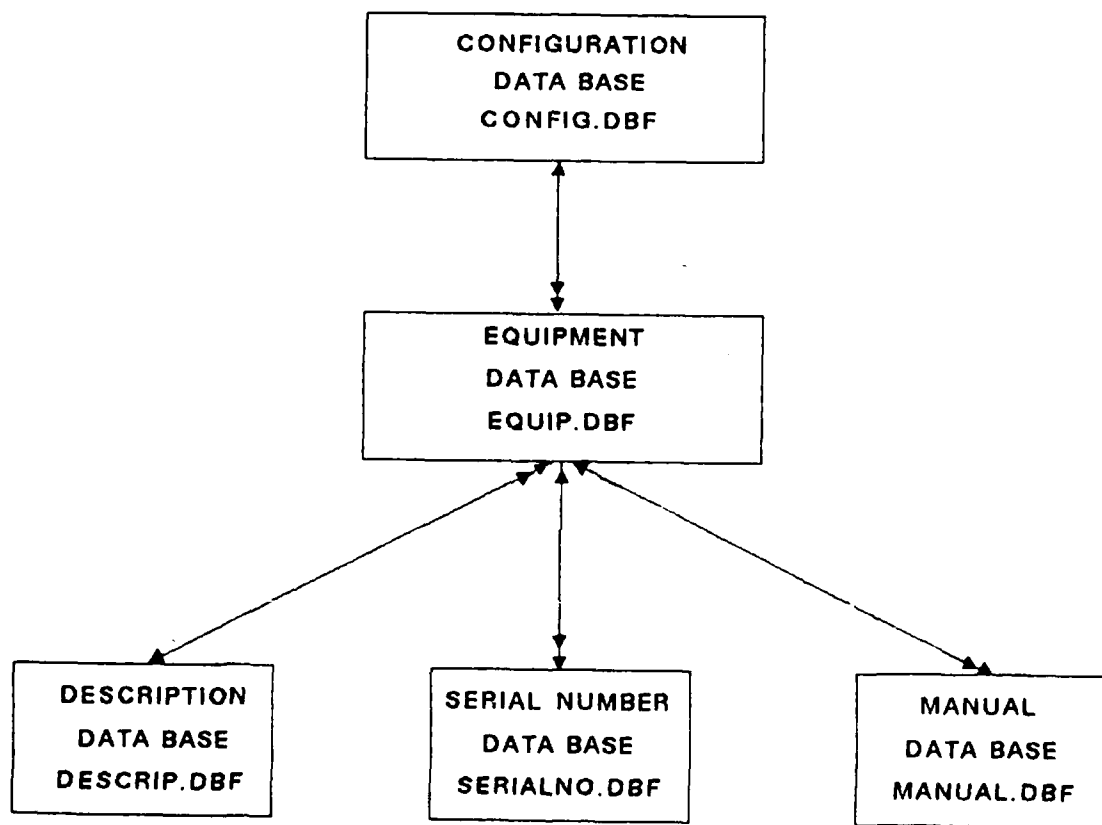
APPENDIX B: MAINTENANCE MANUAL Page 27

Page 4

COSTS.IN Program Listing

151	67	670101	CNFIG MGT QURY & RPT 1 T/CH0.0	95000.0	0.0
152	68	680101	SFTWRE CTL QRY & RPT 1 T/CH0.0	9000.0	0.0
153	7101	710101	COMPUTER OPERATIONS MAN SET0.0	427.00	0.0
154	7201	720101	SYSTEMS PROGRAMMER MAN SET 0.0	607.0	0.0
155	7301	730101	HARDWARE MANUAL SET 0.0	375.0	0.0
156	7401	740101	PROGRAMMERS REF MAN SET 0.0	437.0	0.0
157	XXXX	39XXXX	TRAINING GROUP I 0.0	268637.0	0.0
158	XXXX	39XXXX	TRAINING GROUP II 0.0	164271.0	0.0
159	XXXX	39XXXX	TRAINING GROUP III 0.0	89655.0	0.0
160	XXXX	39XXXX	TRAINING GROUP IV 0.0	21909.0	0.0
161	XXXX	XXXXXX	OPERATOR TRAINING 0.0	14109.0	0.0
162	XXXX	XXXXXX	HARDWARE OVERVIEW 0.0	7000.0	0.0
163	XXXX	XXXXXX	SYSTEMS RESOURCE MGT 0.0	20000.0	0.0
164	XXXX	XXXXXX	SYSTEMS TUNING AND XRAY 0.0	15000.0	0.0
165	XXXX	XXXXXX	DATA COMMUNICATIONS 0.0	10000.0	0.0
166	XXXX	XXXXXX	TAL 0.0	15000.0	0.0
167	XXXX	XXXXXX	SPLICENET MIGRATION WORKSHPO.0	8000.0	0.0
168	81	810101	PM ON-CALL 0.0	0.0	0.0
169	81	810201	PRVT MAINT FOR PER/CALL SIT0.0	0.0	0.0
170	82	820101	ON-CALL MAINTENANCE 0.0	0.0	0.0
171	83	830101	PER-CALL MAINTENANCE 0.0	0.0	0.0
172	84	840101	EMERGENCY PER-CALL MAINT 160.0	0.0	0.0
173	85	850101	NETWORK ADMN COMP(P/O5502010.0	0.0	0.0
174	89	890100	TPS SIMULATION (P/O 520101)0.0	0.0	0.0
175	90	900101	TPS APPL. INT (P/O 520101) 0.0	0.0	0.0
176	91	910101	TPS NTWK INTFCOMP(P/O5201010.0	0.0	0.0
177	92	920101	DSTB TPS PROC CMP(P/O5201010.0	0.0	0.0
178	93	930101	INTGRTE DDL CMP(P/O 5201010.0	0.0	0.0
179	94	940101	TPS RECOVERY CMP(P/O 5201010.0	0.0	0.0
180	95	950101	ENVISION (P/O 510101) 0.0	0.0	0.0
181	96	960101	CONTRACTOR PERS SUP (P/PERS6533.0	0.0	0.0
182	96	960201	SPLICENET MIGRATION SUPPORT0.0	800.0	0.0
183	97	970101	CNFC MGT DATA & RP(MTH COST0.0	4500.0	0.0
184	98	980101	CONTRACTOR TRAVEL COSTS 0.0	0.0	0.0
185	99	991001	PRE-INST TEST FAC.(SEE NOTE0.0	0.0	0.0
186	99	992001	REMOTE BATCH TERML(SEE NOTE0.0	0.0	0.0
187	99	993001	INTERACTIV TERM ACC SEE NOT0.0	0.0	0.0
188	99	994001	HAND ON TEST FAC SEE NOTE 0.0	0.0	0.0

CONFIGURATION MANAGEMENT SYSTEM
BACHMAN DIAGRAM



dBASE III DATA BASE STRUCTURE DESCRIPTIONS

Structure for database : CONFIG.DBF

Date of last update : 12/21/85

<u>Field</u>	<u>Field Name</u>	<u>Type</u>	<u>Width</u>	<u>Dec</u>
1	SITENO	Character	2	
2	SITENAME	Character	50	
3	SITECO	Character	20	
4	SITENAMEFL	Character	40	
5	SITEADD1	Character	40	
6	SITEADD2	Character	40	
7	SITECITY	Character	40	
8	SITESTATE	Character	2	
9	SITEZIP	Character	10	
10	SITETYPE	Character	4	
11	MAINTOPT	Character	4	
12	MAINTRESP	Character	1	

** Total Record Width in Characters ** 253

Structure for database : DESCRIP.DBF

Date of last update : 12/08/85

<u>Field</u>	<u>Field Name</u>	<u>Type</u>	<u>Width</u>	<u>Dec</u>
1	FEATURENO	Character	6	
2	CLIN	Character	4	
3	DESCRIPT	Character	30	
4	MODELNO	Character	10	
5	FDCMODEL	Character	15	
6	TYPECOMPON	Character	1	
7	BASEMAINT	Numeric	7	2
8	NOTES	Memo	10	

** Total Record Width in Characters ** 83

dBASE III DATA BASE STRUCTURE DESCRIPTIONS (Continued)

Structure for database : EQUIP.DBF

Date of last update : 01/08/86

<u>Field</u>	<u>Field Name</u>	<u>Type</u>	<u>Width</u>	<u>Dec</u>
1	EFFDATE	Character	6	
2	SITENO	Character	2	
3	FEATURENO	Character	6	
4	UNIT_PRICE	Numeric	11	2
5	MO_MAINT	Numeric	11	2
6	UNIT_INSTA	Numeric	8	2
7	QTY	Numeric	3	

** Total Record Width in Characters ** 47

Structure for database : MANUAL.DBF

Date of last update : 01/11/86

<u>Field</u>	<u>Field Name</u>	<u>Type</u>	<u>Width</u>	<u>Dec</u>
1	SITENO	Character	2	
2	FEATURENO	Character	6	
3	MANLDESC	Character	24	

** Total Record Width in Characters ** 32

Structure for database : SERIALNO.DBF

Date of last update : 01/08/86

<u>Field</u>	<u>Field Name</u>	<u>Type</u>	<u>Width</u>	<u>Dec</u>
1	EFFDATE	Character	6	
2	SITENO	Character	2	
3	FEATURENO	Character	6	
4	QTY	Numeric	3	
5	TOTQTY	Numeric	3	
6	SERIALNO	Character	8	

** Total Record Width in Characters ** 28

dBASE III DATA BASE STRUCTURE DESCRIPTIONS (Continued)

Structure for database : TED.DBF

Date of last update : 07/18/85

<u>Field</u>	<u>Field Name</u>	<u>Type</u>	<u>Width</u>	<u>Dec</u>
1	FILLER1	Character	1	
2	SITENO	Character	2	
3	CLIN	Character	4	
4	FILLER2	Character	4	
5	FEATURENO	Character	6	
6	FILLER3	Character	6	
7	DESCRIPT	Character	24	
8	FILLER4	Character	1	
9	QTY	Numeric	3	
10	FILLER5	Character	1	
11	UNIT_PRICE	Numeric	11	2
12	FILLER6	Character	1	
13	TOT_PRICE	Numeric	11	2
14	FILLER7	Character	1	
15	MO_MAINT	Numeric	11	2
16	FILLER8	Character	1	
17	MAINT_FAC	Numeric	6	3
18	FILLER9	Character	1	
19	MAINT_MOS	Numeric	6	
20	FILLER10	Character	1	
21	TOT_MAINT	Numeric	10	2
22	FILLER11	Character	1	
23	UNIT_INSTA	Numeric	8	2
24	FILLER12	Character	1	
25	TOT_INSTAL	Numeric	10	2
26	FILLER13	Character	1	
27	COMP_DT_CR	Numeric	11	2
28	FILLER14	Character	1	
29	SYS_DT_CR	Numeric	11	2

** Total Record Width in Characters ** 156

dBASE III Configuration Management System

INDICES COMPOSITION

<u>DATA BASE FILE NAME</u>	<u>INDEX NAME</u>	<u>INDEX KEY COMPOSITION</u>
CONFIG.DBF	CONFIG.NDX	SITENO
DESCRIP.DBF	DESCRIP.NDX	FEATURENO
EQUIP.DBF	EQUIPSIT.NDX	SITENO
	EFEAT.NDX	FEATURENO
	EQUIPSD.NDX	SITENO + EFFDATE
	EQUIPDAT.NDX	SITENO + FEATURENO
	EQUIPPRJ.NDX	EFFDATE + SITENO + FEATURENO
MANUAL.DBF	MANULSIT.NDX	SITENO + FEATURENO
SERIALNO.DBF	SERNOSIT.NDX	SITENO
	SERNODAT.NDX	SITENO + EFFDATE
	SERNOFEA.NDX	SITENO + FEATURENO
	SERNOPRJ.NDX	EFFDATE + SITENO + FEATURENO

dBASE III Configuration Management System

PROGRAM INVOCATION SEQUENCES

<u>DATA LOAD</u>	<u>EQUIPMENT FILE MAINTENANCE</u>	<u>DESCRIPTION FILE MAINTENANCE</u>
SELECTOR.PRG	SELECTOR.PRG	SELECTOR.PRG
MAINMENU.PRG	MAINMENU.PRG	MAINMENU.PRG
NEWDOCMD.PRG	EQUIPCMD.PRG	DESPMOD.PRG
NEWDOCVT.PRG	EQUIPUPD.PRG	DESPUPD.PRG
NEWDOADD.PRG	EQUIPREV.PRG	DESPPREV.PRG
SERNOBLD.PRG		
<u>CONFIGURATION FILE MAINTENANCE</u>	<u>MANUAL FILE MAINTENANCE</u>	<u>SERIAL NUMBER FILE MAINTENANCE</u>
SELECTOR.PRG	SELECTOR.PRG	SELECTOR.PRG
MAINMENU.PRG	MAINMENU.PRG	MAINMENU.PRG
CONFMOD.PRG	MANULCMD.PRG	SERNOCMD.PRG
CONFUPD.PRG	MANULADD.PRG	SERNOUPD.PRG
CONFREV.PRG	MANULUPD.PRG	SERNOREV.PRG
	MANULDEL.PRG	
	MANULREV.PRG	
<u>PROJECT LEVEL REPORTS</u>	<u>SITE LEVEL REPORTS</u>	<u>EFFECTIVE DATE LEVEL REPORTS</u>
SELECTOR.PRG	SELECTOR.PRG	SELECTOR.PRG
MAINMENU.PRG	MAINMENU.PRG	MAINMENU.PRG
REPORCMD.PRG	REPORCMD.PRG	REPORCMD.PRG
PROJRPTS.PRG	SITERPTS.PRG	DATERPTS.PRG
EQPPJRPT.PRG	EQPSTRPT.PRG	EQPDTPRC.PRG
SNOPJRPT.PRG	MNLSTRPT.PRG	EQPDTPNPC.PRG
	SNOSTRPT.PRG	SNODTRPT.PRG
<u>MAINTENANCE DELIVERY ORDER</u>	<u>LABEL GENERATION</u>	
SELECTOR.PRG	SELECTOR.PRG	
MAINMENU.PRG	MAINMENU.PRG	
MAINTDO.PRG	MKLABELS.PRG	

Page 1

SPLICE.PAS Program Listing

```

1 Program SPLICE_CONFIGURER (Textin, Input, Output);
2
3 { ***** }
4 { Title   : SPLICE_Configurer }
5 { Authors : LCDR Robert L. Beard, III, SC, USN }
6 {         : LCDR Winston H. Buckley, SC, USN }
7 {         : LCDR Edward J. Case, SC, USN }
8 { Purpose : To be used by Naval Supply Systems Command, SUP 0473, }
9 {         : personnel as the principal means to configure new Stock }
10 {        : Point Logistic Integrated Communications Environment }
11 {        : (SPLICE) sites. In later versions additions will be }
12 {        : made to assist in preparing augmentations to existing }
13 {        : sites, as well as prepare annual renewal delivery orders }
14 {        : for existing sites }
15 { }
16 { Developed: 04 October 1985 }
17 { Updated  : 07 December 1985 }
18 { ***** }
19
20 { ***** }
21 { General Comments: This program is being designed as an "expert" }
22 { system. It will use a series of "rules of thumb" to develop and }
23 { maintain SPLICE configurations at 62 sites throughout the world. }
24 { The SPLICE configurations developed to date have been done by hand }
25 { and have required extensive "hand message" by technical, financial, }
26 { and contractor personnel to ensure their accuracy. This has proven }
27 { to be both costly in terms of dollars and manpower. By prompting }
28 { the user for key information, this "expert system" will develop }
29 { technically accurate configurations, cost them out, and prepare the }
30 { final delivery orders. }
31 { ***** }
32
33 { The following constants, type and variable declarations are used by the }
34 { Software Bottling Company of New York screen generation program "SCREEN }
35 { SCULPTOR". }
36
37
38 Type
39   STR2 = STRING[2]; STR80 = STRING[80]; STR79 = STRING[79];
40   resSS = (staySS, prevSS, exitSS, nextSS);
41
42 Const CopyrightSS='(C)Copyright 1984, The Software Bottling Company Of New York';
43 { DO NOT REMOVE The Above Copyright Notice
44   This Program may not be used without the above Copyright Notice }
45
46 Const
47   { Esc, Up Arrow Key, Left Arrow Key, Page Up Key }
48   escSS=#27;    uSS='H';    lSS='K';    puSS='I';
49   { Blank, Down Arrow Key, Right Arrow Key, Page Down Key }
50   blankSS=' ';  dSS='P';    rSS='M';    pdSS='O';

```

Page 2

SPLICE.PAS Program Listing

```

51      { Function keys F1-F10 }
52      f1SS='';      f2SS='<';      f3SS='=';      f4SS='>';      f5SS='?';
53      f6SS='@';      f7SS='A';      f8SS='B';      f9SS='C';      f10SS='D';
54      retSS : STR2='';
55
56
57  Var
58      answerSS : String [1];
59      rangeSS : STR80;
60      BeepOnSS, last_fieldSS, retrieveSS : BOOLEAN;
61      actionSS, last_field_actionSS : resSS;
62      hiSS, loSS : REAL;
63      vtypeSS, screenSS, screen_fieldSS, varSS : INTEGER;
64
65
66  { The following constants, type and variable declarations are used by the
67    SPLICE configurer. }
68
69  Type
70      Op_Mode = (Hard, Soft, Document, Train, Maint, Other);
71                { Defines major components categories }
72      Title    = String [19];
73      Names    = Array [1..12] of String [9];
74      CostType = Record
75          { Record for cost data array }
76          featureno : String [6];      { contract feature number }
77          clin      : String [6];      { contract line item number }
78          descript  : String [27];     { contract item description }
79          momaint   : Real;             { monthly maintenance w/ escalation }
80          purchprice: Real;            { purchase price w/ discounts }
81          instcost  : Real;            { installation cost w/ escalation }
82          basemaint : Real;            { basic monthly maintenance cost }
83      End;
84
85      SiteType = Record
86          { Record for site specific information }
87          siteno    : Integer;          { Site number }
88          sitename  : String [27];     { Site name }
89          documentation : Integer;    { Documentation class required }
90          training  : Integer;          { Training class required }
91          maint_options : String [4];  { Currently not used }
92          maint_response : String [1]; { Currently not used }
93          site_type  : String [1];     { Type=MAPS site [M] or Stock Point [S] }
94          site_inst_cost : Real;       { Site installation cost w/o escalation }
95      End;
96
97  Const
98      File1 = 'Costs.IN';              { Name of cost data file }
99      File2 = 'Config.SIT';            { Name of site configuration file }
100     File3 = 'Splice.SCR';             { Name of screen image file }

```

```

101 Month_Name : Names = ('January ', 'February ', 'March ', 'April ',
102                        'May ', 'June ', 'July ', 'August ',
103                        'September', 'October ', 'November ', 'December ');
104
105
106 Var
107     Mode      : Op_Mode;           { Subscript for Totals }
108     SiteInfo   : SiteType;         { Record containing site specific info }
109     Subtotals   : Array [0..5] of Array [1..3] of Real;
110                        { Three subtotals for each section }
111     Totals     : Array [0..5] of Array [1..2] of Real;
112                        { OPN & OMN Totals for each section }
113     CostTable  : Array [1..200] of CostType;
114                        { Array of updated COSTS.IN file info }
115     CardRdr, LIU, Processors, THYPERchannels : Integer;
116     Maint_Months, NETEX_Months, DDN_SW_Months : Integer;
117     A140, A150, A220, A400, A510, AXXX, I, Quantity : Integer;
118     System_Downtime_Component, Downtime_Credit, Maint_Factor : Real;
119     Emerg_Maint_Rate, Extended_Price, Momaint_Esc_Rate : Real;
120     Stock_Point : Char;           { Variables for character responses }
121     Screenfile  : File;           { File of Screen Images }
122     Site_Preps  : String [1];     { Yes or No user response variable }
123     Day         : String [2];     { Effective Day of Delivery Order }
124     Year        : String [4];     { Effective Year of Delivery Order }
125     Line_Number : String [6];     { Contract Line Item Number }
126     Month       : String [9];     { Effective Month of Delivery Order }
127     PRN_File_Name : String [12];  { Output LOTUS .PRN file }
128     Diskfile    : Text;           { Output Delivery Order File }
129
130 { $V-,C-,R- } { Pascal Directives used by SCREEN SCULPTOR. See Compiler Manual }
131 { $I SPLICE1.PAS Include Procedures In This File by SCREEN SCULPTOR. }
132 { SCREEN SCULPTOR(C)
133 (C) COPYRIGHT, THE SOFTWARE BOTTLING COMPANY OF NEW YORK, 1984, 1985
134 ** Turbo Pascal Version, Trade Mark Of Borland International }
135
136 TYPE
137     RECPACKSS = record
138         AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags: INTEGER;
139     end;
140
141 VAR regss : RECPACKSS;
142
143
144 TYPE
145     video_pointerSS = array[1..3840] of CHAR;
146
147 VAR
148 { Video Variables Set By SET_VIDEO.TYPE procedure }
149     vcolorSS, voffSS, vonSS: byte;
150     vdistSS: INTEGER;

```

```

151I     videoSS: ^video_pointerSS;
152I
153I PROCEDURE BEEP(BeepOn: BOOLEAN);
154I BEGIN
155I   if BeepOn then write(chr(7));
156I END;
157I
158I
159I PROCEDURE COLOR(foregr,backgr: BYTE);
160I { Select current color by setting Foreground and Background
161I   Any values between 0 and 15 are acceptable. See Tech Ref Manual
162I }
163I BEGIN
164I   if backgr>7 then foregr:=foregr+16;
165I   TextColor(foregr);
166I   TextBackground(backgr);
167I END; { COLOR }
168I
169I
170I PROCEDURE WRITEC(vtext: STR80);
171I BEGIN
172I   write(vtext);
173I END; { WRITEC }
174I
175I
176I PROCEDURE CLEAR_KBD;
177I { Clear Type Ahead Characters From Keyboard }
178I VAR kchar: CHAR;
179I BEGIN
180I   while keypressed do read(kbd,kchar);
181I END; { CLEAR_KBD }
182I
183I
184I FUNCTION SET_MONITOR_TYPE: INTEGER;
185I { Determine The Type Of Monitor Being Used }
186I VAR j : INTEGER;
187I
188I PROCEDURE CURSOR_SET;
189I { Set Cursor Size }
190I VAR v1,v2,v3 : INTEGER;
191I BEGIN
192I   if j=2 then
193I     begin
194I       v1:=$3d4;
195I       v2:=$3d5;
196I       v3:=$3d9
197I     end
198I   else
199I     begin
200I       v1:=$3b4;

```

APPENDIX B: MAINTENANCE MANUAL Page 38

Page 5

SPLICE.PAS-include file SPLICE1.PAS Program Listing

```

201I          v2:=$3b5;
202I          v3:=$3b9
203I      end;
204I      if (j=2) or (j=3) then
205I      begin
206I          port[v1]:=$0A; port[v2]:=0; { Set High Cursor Scan Line }
207I          port[v1]:=$0B; port[v2]:=7; { Set Low Cursor Scan Line }
208I          port[v3]:=1;           { Set Border Color to BLUE }
209I      end;
210I      END; { CURSOR_SET }
211I
212I BEGIN
213I      j:=mem[$40:$10]; { Figure out the monitor type }
214I      j:=(j and $0030) DIV 16;
215I      CASE j OF
216I          0: begin writeln('Illegal Monitor Mode'); halt end;
217I          1: begin { Set 40 column color to 80 column color }
218I              writeln('Use MODE command to set to 80. ( MODE CO80 )'); halt
219I          end;
220I          2: videoSS:=ptr($b800,0); { Graphics 80 }
221I          3: videoSS:=ptr($b000,0); { Monochrome }
222I      END;
223I      voffSS:=$1; vonSS:=$29; vdispSS:=$3d8; { Video Off, On, Location }
224I      CURSOR_SET; { Set To A Large Cursor }
225I      COLOR(14,1); { Set Default Color }
226I      SET_MONITOR_TYPE:=j;
227I      END; { SET_MONITOR_TYPE }
228I
229I
230I PROCEDURE DISPLAY_SCREEN (var screenfile : FILE);
231I { Load Screen From Disk. Display To Monitor }
232I VAR bload: array[1..3968] of CHAR;
233I      exist : Boolean;
234I
235I      PROCEDURE VIDEO_OFF; { Turn Video Off }
236I          BEGIN port[vdispSS]:=voffSS; END;
237I
238I      PROCEDURE VIDEO_ON; { Turn Video On }
239I          BEGIN port[vdispSS]:=vonSS; END;
240I BEGIN
241I      if IOresult=0 then
242I          begin
243I              exist:=TRUE;
244I              blockread (screenfile, bload[1], 31);
245I              VIDEO_OFF;
246I              move (bload[8], videoSS, 3840);
247I              VIDEO_ON;
248I          end
249I      else exist:=FALSE;
250I      if not exist then

```



```

251I      begin
252I          color (15, 4);
253I          gotoxy (25, 13);
254I          write ('G, 'Part of SPLICE.SCR is missing. ');
255I      end;
256I      retSS := '';
257I  END; { DISPLAY_SCREEN }
258I
259I { See SCREEN SCULPTOR Manual For A Description Of GETITEM }
260I  PROCEDURE GETITEM(
261I      COL,LIN,LEN :      BYTE;
262I      ITYPE :          CHAR;
263I      VAR WITEM :      STR80;
264I      PICT :          STR80;
265I      ITEM_LOW,ITEM_HIGH : STR80;
266I      VAR RET :        STR2;
267I      RETRIEVE :      BOOLEAN;
268I      FGR_COLOR,BGR_COLOR : BYTE
269I  );
270I
271I  TYPE
272I      PICT_TYPE = set of CHAR;
273I
274I  CONST
275I      confirm=FALSE; { If FALSE auto-skip to next field when field is filled }
276I      l='K'; r='M'; u='H'; d='P'; dl='S'; ins='R'; pu='I'; pd='Q';
277I      { Define The Function Keys }
278I      f1=''; f2='<'; f3='='; f4='>'; f5='?';
279I      f6='@'; f7='A'; f8='B'; f9='C'; f10='D';
280I      special_keys: PICT_TYPE = [l,r,u,d,dl,ins,pu,pd];
281I      pict_elements: PICT_TYPE = ['X','U','L',' ','#','9','8'];
282I      bk: BYTE=8; esc: BYTE=27; cr: BYTE=13;
283I
284I  VAR
285I      hcol,pcol,tcol,pict_dec,item_dec,tempb1,tempb2,plen,ilen: BYTE;
286I      kchar: str2; range_check,clear25: BOOLEAN;
287I      check,end_of_field,begin_of_field,sign_flag,
288I      special_dec_flag,valid_char: BOOLEAN;
289I      temp_item, item: STR80;
290I      fchar: CHAR;
291I
292I  FUNCTION DATE_CHECK(datevar: STR80): BOOLEAN;
293I  { Checks For Date Validity Excluding the following:
294I    Does not check Leap Years. If datevar is correct then DATE_CHECK is TRUE }
295I  CONST
296I      month_days: array[1..12] of INTEGER=(31,28,31,30,31,30,31,31,30,31,30,31);
297I  VAR mm,dd,yy: STR2;
298I      mmi,ddi,yyi: INTEGER;
299I      error: INTEGER;
300I      ch_date: BOOLEAN;

```

```

301I BEGIN
302I if ord(datevar[0])<>8 then
303I   DATE_CHECK:=FALSE
304I else
305I begin
306I   ch_date:=TRUE;
307I   mm:=copy(datevar,1,2);
308I   dd:=copy(datevar,4,2);
309I   yy:=copy(datevar,7,2);
310I   val(mm,mmi,error);
311I   if (error<>0) or (mmi<1) or (mmi>12) then ch_date:=FALSE;
312I   if ch_date then
313I   begin
314I     val(dd,ddi,error);
315I     if (error<>0) or (ddi<1) or (ddi>month_days(mmi)) then ch_date:=FALSE;
316I   end;
317I   if ch_date then
318I   begin
319I     val(yy,yyi,error);
320I     if error<>0 then ch_date:=FALSE;
321I   end;
322I   DATE_CHECK:=ch_date;
323I end;
324I END; { PROCEDURE DATE_CHECK }
325I
326I FUNCTION CHECK_DATE(DATE, DATE_LOW, DATE_HIGH: STR80): BOOLEAN;
327I { Check Validity If Date and whether it falls between low and high }
328I { If low range date is higher than high range date then we assume }
329I { we crossed centuries eg. 09/09/84 to 01/01/10 }
330I { Also a null date is ignored }
331I CONST dnull = ' / / ';
332I VAR   ch_date: BOOLEAN;
333I BEGIN
334I if date<>dnull then ch_date:=DATE_CHECK(date) else ch_date:=TRUE;
335I if ch_date and (date<>dnull) and (date_low<>dnull) and (date_high<>dnull) then
336I begin
337I   if ch_date then ch_date:=DATE_CHECK(date_low);
338I   if ch_date then ch_date:=DATE_CHECK(date_high);
339I   if ch_date then
340I   begin
341I     date:=copy(date,7,2)+copy(date,1,6);
342I     date_low:=copy(date_low,7,2)+copy(date_low,1,6);
343I     date_high:=copy(date_high,7,2)+copy(date_high,1,6);
344I     if (date_low=date_high) then { Low Date = High Date }
345I     begin
346I       if (date<date_low) or (date>date_high) then ch_date:=FALSE;
347I     end else { Low Date < High Date }
348I       if (date<date_low) and (date>date_high) then ch_date:=FALSE;
349I   end;
350I end;

```

```

351I if ch_date then CHECK_DATE:=TRUE else begin CHECK_DATE:=FALSE; end;
352I END; {PROCEDURE CHECK_DATE}
353I
354I FUNCTION CHECK_RANGE(VAR item, item_low, item_high: STR80): BOOLEAN;
355I { Check to see whether item is within and including low and high }
356I VAR itemr, lowr, highr: REAL;
357I     errori, errorl, errorh: INTEGER;
358I BEGIN
359I CHECK_RANGE:=TRUE;
360I val(item_low,lowr,errorl);
361I val(item_high,highr,errorh);
362I val(item,itemr,errori);
363I if (errorl=0) and (errorh=0) and (errori=0) then
364I begin
365I   if itemr<lowr then CHECK_RANGE:=FALSE
366I   else if itemr>highr then CHECK_RANGE:=FALSE;
367I end else
368I   CHECK_RANGE:=FALSE;
369I END; { PROCEDURE CHECK_RANGE }
370I
371I PROCEDURE MESSAGE(mess_num: BYTE);
372I { Displays A Message On Line 25 and sets global clear25 to TRUE }
373I VAR mess, temp_item: STR79; mess_length, start_col: INTEGER;
374I BEGIN
375I   color (14,1); gotoxy (1, 25); clreol;
376I   case mess_num of
377I     1: mess:=' Only 0 thru 9 Allowed ';
378I     2: mess:=' Only 0 thru 9 or a space Allowed ';
379I     3: mess:=' BAD Date OR Not Within '+item_low+' & '+
380I       item_high+'. Use [Del] To Blank Out Digits. ';
381I     4: mess:=' Number Not Within '+item_low+' & '+item_high+' Range ';
382I     5: mess:=' Only 0 thru 9, decimal point OR - sign Allowed ';
383I     6: mess:=' Only Y or N Allowed ';
384I     7: mess:=' Only M or F Allowed ';
385I     8: mess:=' No More Room For Digits. Use [Del] key to remove ';
386I     9: mess:=' No Space For Negative Numbers. Input Positions Must Be Larger ';
387I   end; { case }
388I mess_length:=ord(mess[0]);
389I start_col:=(79-mess_length) DIV 2;
390I clear25:=TRUE;
391I gotoxy(start_col,25);
392I COLOR(15, 4);
393I write('^G, mess);
394I gotoxy(hcol,lin);
395I COLOR (14, 1);
396I CLEAR_EBD;
397I END; {MESSAGE PROCEDURE}
398I
399I FUNCTION GETCHAR(ct,y: CHAR; VAR kchar: STR2):BOOLEAN;
400I { If GETCHAR=TRUE on return then kchar= (1 r d u d l in p p l o s e o r 1 1 1)

```

```

401I { if GETCHAR=FALSE on return then kchar is alpha numeric chars }
402I { ctype must be one of the following}
403I { U=Uppercase, L=Lower Case, X=Any Char, 9=0..9, ' ', #=0..9,-,+,.. }
404I { GETCHAR will filter out any control characters }
405I TYPE PICT_TYPE = set of CHAR;
406I CONST esc = 27; cr = 13; bk = 8;
407I     l='K'; r='M'; u='H'; d='P'; dl='S'; ins='R'; pu='I'; pd='Q';
408I     f1='<'; f2='>'; f3='='; f4='>'; f5='?';
409I     f6='@'; f7='A'; f8='B'; f9='C'; f10='D';
410I     special_keys: PICT_TYPE = {l,r,u,d,dl,ins,pu,pd};
411I     func_keys: PICT_TYPE = {f1,f2,f3,f4,f5,f6,f7,f8,f9,f10};
412I var  str: CHAR; special,correct: BOOLEAN;
413I     temps: STR79;
414I BEGIN
415I     kchar:='';
416I     GETCHAR:=TRUE; correct:=FALSE;
417I     repeat { until getchar = TRUE }
418I     special:=TRUE;
419I     repeat { until a valid picture character }
420I     repeat until keypressed;
421I     read(kbd,kchar[1]);
422I     if keypressed and (kchar[1]=chr(esc)) then
423I     begin
424I         read(kbd,kchar[2]);
425I         kchar[1]:=chr(0);
426I         kchar[0]:=chr(2);
427I     end else
428I         kchar[0]:=chr(1);
429I { Clear Line 25 }
430I     if clear25 then
431I     begin
432I         color (14, 1);
433I         gotoxy (1,25);
434I         clreol;
435I         gotoxy(hcol,lin);
436I         clear25:=FALSE;
437I         color (FGR_COLOR, BGR_COLOR);
438I     end;{ Clear Line }
439I     if (not (ord(kchar[1]) in {esc,cr,bk})) and (ord(kchar[0])=1) then
440I     begin
441I         str:=kchar[1];
442I         if (str>=' ') and (str<=' ') then
443I         case ctype of
444I             'X': correct:=TRUE;
445I             'U': begin
446I                 if str in ['a'..'z'] then str:=chr(ord(str) and 54);
447I                 kchar[1]:=str; correct:=TRUE;
448I             end;
449I             'L': begin
450I                 if str in ['A'..'Z'] then str:=chr(ord(str) or $20);

```

```

451I      kchar[1]:=str; correct:=TRUE;
452I      end;
453I      '#': if (str in ['0'..'9','-','.']) then correct:=TRUE else message(5);
454I      '9': if str in ['0'..'9',' '] then correct:=TRUE else message(2);
455I      '8': if str in ['0'..'9'] then correct:=TRUE else message(1);
456I      end ( case )
457I      end ( begin )
458I      else
459I      begin (special character)
460I      GETCHAR:= FALSE;
461I      correct:=TRUE;
462I      str:=kchar[1];
463I      end;
464I      until correct;
465I      if (ord(kchar[0])=2) then ( see if it is a special character )
466I      begin
467I      special:=FALSE;
468I      GETCHAR:=TRUE;
469I      if (kchar[2] in special_keys) or (kchar[2] in func_keys) then
470I      begin
471I      GETCHAR:=FALSE;
472I      special:=TRUE;
473I      end else BEEP(BeepOnSS);
474I      end;
475I      until special;
476I      ret:=kchar;
477I      END; ( GETCHAR FUNCTION )
478I
479I      PROCEDURE DECH; ( Positions Cursor At the Next Non Edit Character )
480I      VAR elem_end: BOOLEAN; tempbl: BYTE;
481I      BEGIN
482I      if hcol<>(col+tccl-1) then
483I      begin
484I      tempbl:=pcol;
485I      elem_end:=FALSE;
486I      repeat
487I      tempbl:=tempbl-1;
488I      if (pict[tempbl] in pict_elements) or (tempbl<1) then elem_end:=TRUE;
489I      until elem_end;
490I      if tempbl>=1 then
491I      begin
492I      hcol:=hcol-(pcol-tempbl);
493I      pcol:=tempbl;
494I      end;
495I      end else
496I      begin_of_field:=TRUE;
497I      END; ( DECH PROCEDURE )
498I
499I      PROCEDURE INCH; ( Positions Cursor At the Next Non Edit Character )
500I      VAR elem_end: BOOLEAN; tempbl: BYTE;

```

Page 11

SPLICE.PAS-include file SPLICE1.PAS Program Listing

```

501I BEGIN
502I if hcol<>(col+len-1) then
503I begin
504I   tempbl:=1;
505I   elem_end:=FALSE;
506I   repeat
507I     tempbl:=tempbl+1;
508I     if (pict[pcol+tempbl-1] in pict_elements) or ((pcol+tempbl)>(len)) then
509I       elem_end:=TRUE;
510I   until elem_end;
511I   if tempbl<=(len) then
512I     begin
513I       hcol:=hcol+tempbl-1;
514I       pcol:=pcol+tempbl-1;
515I     end;
516I   end else
517I   end_of_field:=TRUE;
518I END; { INCH PROCEDURE }
519I
520I PROCEDURE STRIP_BLANKS(VAR temp_item: STR80);
521I { Strip Blanks On Both Sides Of passed item }
522I VAR i,j: BYTE;
523I BEGIN
524I if temp_item<>' ' then
525I begin
526I   j:=ord(temp_item[0]);
527I   { Strip Leading Blanks }
528I   i:=0;
529I   while (temp_item[i+1]=' ') and (i<j) do i:=i+1;
530I   if (i>0) and (i<j) then temp_item:=copy(temp_item,i+1,j-i)
531I   else if (i=j) and (temp_item[j]=' ') then temp_item:='';
532I   i:=pos(' ',temp_item); { strip trailing blanks }
533I   if i>0 then temp_item:=copy(temp_item,1,i-1);
534I end;
535I END; { STRIP_BLANKS PROCEDURE }
536I
537I BEGIN { Main Procedure Of GETITEM }
538I item:=witem; { Store Actual Item In A Work Variable }
539I clear25:=FALSE;
540I if itype='D' then
541I   begin
542I     pict:='88/88/88';
543I     len:=8;
544I   end;
545I if itype='Y' then
546I begin
547I   if not (item[1] in ['Y','N']) then item:='Y';
548I   pict:='U';
549I   len:=1;
550I end;

```

```

551I if itype='M' then
552I begin
553I   if not (item[1] in ['M','F']) then item:='M';
554I   pict:='U';
555I   len:=1;
556I end;
557I end_of_field:=FALSE;
558I begin_of_field:=FALSE;
559I if (pict='') and (itype='C') then pict:='X';
560I plen:=ord(pict[0]);
561I fchar:=pict[plen];
562I ilen:=ord(item[0]);
563I (* Fill Item with blanks *)
564I if itype<>'N' then (* If item is non numeric *)
565I begin
566I   while ilen<len do
567I     begin
568I       item:=item+' ';
569I       ilen:=ilen+1;
570I     end;
571I   while plen<len do
572I     begin
573I       pict:=pict+fchar;
574I       plen:=plen+1;
575I     end;
576I end else (* If item is numeric *)
577I begin
578I   strip_blanks(item);
579I   if item='' then item:='0';
580I   ilen:=ord(item[0]);
581I   while ilen<len do
582I     begin
583I       item:=' '+item;
584I       ilen:=ilen+1;
585I     end;
586I   while plen<len do
587I     begin
588I       pict:='#'+pict;
589I       plen:=plen+1;
590I     end;
591I   if ord(pict[0])>len then pict:=copy(pict,ord(pict[0])-len+1,len);
592I   if ord(item[0])>len then item:=copy(item,1,len);
593I   ilen:=ord(item[0]); plen:=ord(pict[0]);
594I   pict_dec:=pos('.',pict);
595I   item_dec:=pos('.',item);
596I (* Align Decimal Positions If Necessary *)
597I if pict_dec<>item_dec then
598I begin (* alignment *)
599I   check:=TRUE;
600I (* If picture has no decimal point and item does)

```

```

601I if (pict_dec=0) and check then
602I begin
603I   item:=copy(item,1,item_dec-1);
604I   fillchar(temp_item,ord(pict[0])-ord(item[0]),' ');
605I   item:=temp_item+item;
606I   check:=FALSE;
607I end;
608I { If item has no decimal point and pict does}
609I if (item_dec=0) and check then
610I begin
611I   strip_blanks(item);
612I   tempb2:=plen-pict_dec; { # of decimal points };
613I   fillchar(temp_item,tempb2,item[ord(item[0])]);
614I   item:=item+'.'+temp_item; { Add decimal trailing digits }
615I   ilen:=ord(item[0]); { Get length of item }
616I   while ilen<plen do { Add blanks left}
617I     begin
618I       item:=' '+item;
619I       ilen:=ilen+1;
620I     end;
621I   if ilen>plen then { If The Item > Picture }
622I     begin
623I       item:=copy(item,1,pict_dec-1);
624I       item:=item+'.'+temp_item;
625I     end;
626I     check:=FALSE;
627I end;
628I { If item decimal is further right than pict dec}
629I if (item_dec>pict_dec) and check then
630I begin { Move the item to the left dropping off numbers  pict's}
631I   plen:=ord(pict[0]);
632I   ilen:=ord(item[0]);
633I   item:=copy(item,item_dec-pict_dec+1,ilen-(item_dec-pict_dec));
634I   ilen:=ord(item[0]);
635I   tempb1:=plen-ord(item[0]);
636I   fillchar(temp_item,tempb1,item[ilen]);
637I   item:=item+temp_item;
638I   ilen:=ord(item[0]);
639I   while ilen<plen do { Add blanks left}
640I     begin
641I       item:=' '+item;
642I       ilen:=ilen+1;
643I     end;
644I   check:=FALSE;
645I end;
646I { If pict decimal is further right than item's}
647I if (pict_dec>item_dec) and check then
648I begin
649I   tempb2:=plen-pict_dec;
650I   item:=copy(item,1,item_dec+tempb2);

```



```

651I   ilen:=ord(item[0]);
652I   while ilen<len do
653I       begin
654I           item:=' '+item;
655I           ilen:=ilen+1;
656I       end;
657I   check:=FALSE;
658I end;
659I end { alignment };
660I end { fillings};
661I (* Copy edit characters to item *)
662I   for tempb1:=1 to len do
663I       if not (pict[tempb1] in pict_elements) then item[tempb1]:=pict[tempb1];
664I (* Display The item on the screen *)
665I   color(FGR_COLOR, BGR_COLOR);
666I   gotoxy(col,lin);
667I   writec(item);
668I (* Get Data From Screen If Retrieve is True)
669I if retrieve then
670I begin { Retrieve }
671I (* Move cursor to first position by bypassing edit chars )
672I   pcol:=1;
673I   while (not (pict[pcol] in pict_elements)) and (pcol<=len) do pcol:=pcol+1;
674I (* Readjust column )
675I   tcol:=pcol;
676I (* Handle Non Numeric Type Of Item *)
677I if (itype<>'N') and (pcol<=len) then
678I { pcol is position of cursor within field}
679I begin (* Non Numeric Field *)
680I   repeat { Until range_check = TRUE }
681I       pcol:=tcol;
682I       hcol:=col+pcol-1;
683I       gotoxy(hcol,lin); (* Go to location on screen*)
684I       repeat
685I           end_of_field:=FALSE;
686I           begin_of_field:=FALSE;
687I           special:=FALSE;
688I           if getchar(pict[pcol],kchar) then
689I               begin
690I                   writec(kchar);
691I                   item[pcol]:=kchar[1];
692I                   inch;
693I                   gotoxy(hcol,lin);
694I               end else
695I                   special:=TRUE;
696I               if special then
697I                   begin { Special Key Pressed }
698I                       ret:=kchar;
699I                       special:=FALSE;
700I                       if kchar[1]=chr(bk) then { It is backspace }

```

```

701I      begin
702I          dech;
703I          gotoxy(hcol,lin); {Left}
704I      end else
705I      if (ord(kchar[0])=2) and (kchar[2] in [1,r,dl,ins]) then
706I      begin
707I          case kchar[2] of
708I              l: begin dech; gotoxy(hcol,lin); end; {Left}
709I              r: begin inch; gotoxy(hcol,lin); end; {Right}
710I              dl: begin {Delete}
711I                  tempb2:=pcol+1; {Find where the next edit char starts}
712I                  while (pict[tempb2] in pict_elements) and (tempb2<=len) do
713I                      { tempb1=start, tempb2:=end}
714I                      tempb2:=tempb2+1;
715I                      tempb2:=tempb2-1;
716I                      for tempb1:=pcol to tempb2-1 do {move chars left}
717I                          begin { & put blank at end}
718I                              item[tempb1]:=item[tempb1+1];
719I                          end;
720I                      item[tempb2]:=' ';
721I                      {rewrite the item}
722I                      gotoxy(col,lin);
723I                      writec(item);
724I                      gotoxy(hcol,lin);
725I                  end;
726I              ins: begin {Insert}
727I                  tempb2:=pcol+1;
728I                  while (pict[tempb2] in pict_elements) and (tempb2<=len) do
729I                      tempb2:=tempb2+1;
730I                      tempb2:=tempb2-1;
731I                      for tempb1:=tempb2 downto pcol+1 do
732I                          begin
733I                              item[tempb1]:=item[tempb1-1];
734I                          end;
735I                      item[pcol]:=' ';
736I                      gotoxy(col,lin);
737I                      writec(item);
738I                      gotoxy(hcol,lin);
739I                  end;
740I              end { Case kchar };
741I      end
742I      else {esc,cr,pjup,pjdn,up,dn}
743I      special:=TRUE;
744I      end {If backspace };
745I      if end_of_field or begin_of_field then BEEP(BeepOnSS);
746I      until (end_of_field and (not confirm)) or begin_of_field or special;
747I      tempb1:=len; { Strip Trailing Blanks }
748I      if itype='C' then
749I          while (item[tempb1]=' ') and (tempb1>0) do tempb1:=tempb1-1;
750I      item[0]:=chr(tempb1);

```

```

751I   range_check:=TRUE;
752I   if itype='D' then
753I   begin
754I       range_check:=check_date(item,item_low,item_high);
755I       if not range_check then message(3);
756I   end;
757I   if itype='Y' then
758I       if not (item[1] in ['Y','N']) then
759I       begin
760I           range_check:=FALSE;
761I           message(6);
762I       end;
763I   if itype='M' then
764I       if not (item[1] in ['M','F']) then
765I       begin
766I           range_check:=FALSE;
767I           message(7);
768I       end;
769I   until range_check;
770I   end { If non numeric type of item} else { if Numeric }
771I   if (itype='N') then
772I   begin
773I       tcol:=len;
774I       repeat { Until range_check=TRUE }
775I       len:=tcol;
776I       tempbl:=len;
777I       len:=pos('.',item);
778I       range_check:=FALSE;
779I       if len=0 then len:=tempbl
780I       else len:=len-1; { Item has decimal point }
781I       hcol:=col+len-1;
782I       pcol:=len;
783I       gotoxy(hcol,lin);
784I       special:=FALSE;
785I       sign_flag:=FALSE;
786I       end_of_field:=FALSE;
787I       dec_flag:=FALSE;
788I       repeat
789I           valid_char:=FALSE;
790I           if getchar('#',kchar) then
791I           begin { Not Special }
792I               case kchar of
793I               '-' : { Sign } if not sign_flag then valid_char:=TRUE;
794I               '.' : { Decimal point }
795I                   if (len<>tempbl) and (not dec_flag) then
796I                   begin
797I                       hcol:=hcol+2; pcol:=len+2; gotoxy(hcol,lin);
798I                       dec_flag:=TRUE; sign_flag:=TRUE;
799I                   end;
800I               '0'..'9': valid_char:=TRUE;

```

```

801I   end { Case kchar };
802I   { sign_flag = if FALSE we allow minus (-) sign }
803I   { dec_flag = if FALSE we allow decimal (.) point }
804I   if (valid_char) and (not dec_flag) then { Integer Portion }
805I   begin
806I       if (item[1]<>' ') and (len<>tempb1) and (sign_flag) and
807I           not ((ord(item[0])>1) and (item[1]='-') and (item[2]='0')) then
808I           message(8) { Overflow Numeric Field }
809I       else
810I       begin
811I           if (not sign_flag) then { Erase Old Entry. Start New One }
812I               begin
813I                   { Sign Allowed }
814I                   for pcol:=1 to len-1 do item[pcol]:=' ';
815I                   if tempb1>len then
816I                       for pcol:=len+2 to tempb1 do item[pcol]:='0';
817I                       if (kchar[1]<>'0') then sign_flag:=TRUE;
818I                       { Check if field is too small to accomodate a minus sign }
819I                       if kchar[1]='-' then
820I                           begin
821I                               if (len-1)<=0 then
822I                                   begin
823I                                       message(9);
824I                                       sign_flag:=FALSE;
825I                                   end else
826I                                   begin
827I                                       item[len-1]:='-';
828I                                       item[len]:='0';
829I                                   end;
830I                               end else
831I                               item[len]:=kchar[1];
832I
833I                               gotoxy(col,lin);
834I                               writec(item);
835I                               gotoxy(hcol,lin);
836I                           end else
837I                           begin
838I                               { Insert A Digit. No Sign Allowed }
839I                               if not ((item[len]='0') and (item[len-1]='-')) then
840I                                   if not end_of_field then
841I                                       for pcol:=1 to len-1 do item[pcol]:=item[pcol+1];
842I
843I                                       item[len]:=kchar;
844I                                       gotoxy(col,lin);
845I                                       writec(item);
846I                                       gotoxy(hcol,lin);
847I                                   end;
848I                                   if (item[1]<>' ') and (len=tempb1) then end_of_field:=TRUE;
849I                               end;
850I                           end { Integer Portion }

```

```

851I     else ( Decimal Portion )
852I         if valid_char and (sign_flag) then
853I             begin
854I                 item[pcol]:=kchar(1);
855I                 writec(item[pcol]);
856I                 if not end_of_field then
857I                     begin
858I                         hcol:=hcol+1;
859I                         pcol:=pcol+1
860I                     end;
861I                 if pcol>tempbl then
862I                     begin
863I                         hcol:=hcol-1;
864I                         pcol:=pcol-1;
865I                         end_of_field:=TRUE
866I                     end;
867I                 gotoxy(hcol,lin);
868I             end;
869I     end { getchar is FALSE } else { getchar is TRUE }
870I     special:=TRUE;
871I     { Special Keys. DEL}
872I     if special then
873I         begin
874I             ret:=kchar;
875I             special:=FALSE;
876I             if (ord(kchar(0))=2) then
877I                 begin { Case }
878I                     case kchar(2) of
879I                         dl,1: { DELETE KEY PRESSED OR LEFT ARROW KEY }
880I                             case dec_flag of
881I                                 False: { Integer Portion }
882I                                     begin
883I                                         sign_flag:=TRUE;
884I                                         for pcol:=len downto 2 do item[pcol]:=item[pcol-1];
885I                                         if (item[len] in [' ','-']) then
886I                                             begin
887I                                                 item[len]:='0';
888I                                                 sign_flag:=FALSE;
889I                                             end;
890I                                         item[1]:=' ';
891I                                         gotoxy(col,lin);
892I                                         writec(item);
893I                                         gotoxy(hcol,lin);
894I                                         end_of_field:=FALSE;
895I                                     end { F };
896I                                 True: { Decimal Portion }
897I                                     { Put 0 @ Cursor. Check If Going To Integer Part}
898I                                     if pict[pcol-1]='.' then {Are We In Integer Part?}
899I                                         begin {YES. Initialize Variables}
900I                                             hcol:=col+len-1;

```

```

901I      gotoxy(hcol,lin);
902I      dec_flag:=FALSE;
903I      end_of_field:=FALSE;
904I  end else
905I  begin
906I      if not end_of_field then
907I      begin
908I          hcol:=hcol-1;
909I          pcol:=pcol-1
910I      end;
911I      gotoxy(hcol,lin);
912I      item[pcol]:='0';
913I      writec(item[pcol]);
914I      gotoxy(hcol,lin);
915I      end_of_field:=FALSE;
916I  end;
917I  { T }
918I  end { dec_flag CASE };
919I  u,d,l,r,pu,pd,f1,f2,f3,f4,f5,f6,f7,f8,f9,f10: special:=TRUE;
920I  end; { DELETE KEY CASE }
921I  end { Case } else
922I  if (ord(kchar[1]) in [cr, esc]) then special:=TRUE;
923I  end { Special };
924I  if end_of_field and (not special) then BEEP(BeepOnSS);
925I  until special or (end_of_field and (not confirm));
926I
927I  { Get Old Length back and find point position }
928I  len:=tcol;
929I  pcol:=pos('.',pict);
930I
931I  { If no decimal point and 1st position is minus or blank then set to 0 }
932I  if (item[len] in [' ', '-']) and (pcol=0) then
933I  begin
934I      item[len]:='0';
935I      gotoxy(col,lin);
936I      writec(item);
937I  end;
938I
939I  temp_item:=item;
940I  strip_blanks(item);
941I  range_check:=check_range(item,item_low,item_high);
942I  if not range_check then
943I  begin
944I      message(4);
945I      item:=temp_item;
946I  end;
947I
948I  until range_check;
949I  end; { Numeric }
950I  end { Retrieve } else

```

```

951I begin
952I   if itype='N' then strip_blanks(item);
953I   if itype='C' then
954I     begin
955I       tempbl:=len; { Strip Trailing Blanks }
956I       while (item[tempbl]=' ') and (tempbl>0) do tempbl:=tempbl-1;
957I       item[0]:=chr(tempbl);
958I     end;
959I end;
960I witem:=item; { Return result Back To witem }
961I END;{ GETITEM PROCEDURE}
962
963 { *   This is a summary of the procedures in SPLICE1.PAS
964
965 PROCEDURE BEEP(BeepOn: BOOLEAN);           { Sound Beep if BeepOn=TRUE }
966 PROCEDURE CLEAR_KBD;                       { Clear Keyboard Buffer }
967 PROCEDURE COLOR(foregr,backgr:BYTE);       { Set Color }
968 PROCEDURE WRITEC(vtext: STR80);           { Write Chars Using Color }
969 FUNCTION SET_MONITOR_Type: INTEGER;        { Determine Monitor Type }
970 { Display A Screen Sculptor Screen }       { 2=Color, 3=Mono }
971 PROCEDURE DISPLAY_SCREEN(screen_name: STR80; Var file_existSS: BOOLEAN);
972 { Display And Get An Item From Screen. See Detailed Description In Manual }
973 PROCEDURE GETITEM(COL,LIN,LEN :           BYTE;   { Column, Line, Length }
974                   ITYPE :               CHAR;   { Type= C, N, D, Y, M }
975                   Var WITEM :            STR80;  { Variable Name }
976                   PICT :                 STR80;  { Picture X, U, L, 9, 8 # }
977                   ITEM_LOW,ITEM_HIGH :   STR80;  { Range - Numerics/Date Only }
978                   Var RET :              STR2;   { Returned Code }
979                   RETRIEVE :              BOOLEAN; { False=Disp Only, True=Get }
980                   FGR_COLOR,BGR_COLOR :   BYTE   { Colors Foregr, Backgr }
981                   ); EXTERN;
982
983
984 { * * * * *
985
986 GLOBAL PROCEDURES
987
988 { * * * * *
989
990 { Global Procedures used by SCREEN SCULPTOR }
991
992
993 PROCEDURE ACCEPT_INPUTS;
994
995 { Display a prompt on line 25 of the CRT and ask the user if he/she wants
996   to accept or reject the data values input thus far. A "Y" or "N" response
997   only is allowed. }
998
999 Begin { Procedure ACCEPT_INPUTS }
1000   COLOR (14, 1); { Set Foreground & background colors }

```

Page 21

SPLICE.PAS Program Listing

```

1001 GOTOXY (1, 25);           { Position cursor col 1, row 25 }
1002 ClrEol;                  { Clear row 25 with blanks }
1003 WRITE ('      Do you accept the input values thus far?   Yes or No   ');
1004 answerSS := 'N';
1005 GETITEM (70, 25, 1, 'Y', answerSS, 'U', '', '', retSS, True, 12, 1);
1006 GOTOXY (1, 25);           { Position cursor col 1, row 25 }
1007 TextBackground (1);      { Set background color to BLUE }
1008 ClrEol;                  { Clear row 25 with blanks }
1009 End; { Procedure ACCEPT_INPUTS }
1010
1011
1012 PROCEDURE RET_STATUS;
1013 { Check Status Of Variable retSS and return a code in 'actionSS' & set 'varSS'
1014   This procedure is called immediately following GETITEM }
1015
1016 { Input to this procedure:
1017   when retSS is length 1 the values are any of the ASCII chars
1018   when retSS is length 2 the values are uSS, lSS, puSS, pdSS, function keys
1019                                     dSS, rSS
1020                                     { See Const Section For Meanings } }
1021 { Output:
1022   The following codes are returned in actionSS : nextSS, prevSS,
1023                                               exitSS, staySS }
1024 { Based upon 'actionSS' this procedure will then set 'varSS' to an integer,
1025   which represents the next item (variable ) to get. }
1026
1027 Begin
1028   last_field actionSS := exitSS;
1029   actionSS := nextSS;      { Initialize Action Code }
1030   IF retrieveSS THEN      { Is retrieveSS TRUE? }
1031     Begin
1032       IF ord (retSS[0]) = 2 THEN { Is retSS length 2 ? }
1033         Begin
1034           CASE retSS[2] of
1035             { Action to be taken depending on last key pressed }
1036             uSS, lSS : actionSS := prevSS; { Up Key, Left Key }
1037             dSS, rSS : actionSS := nextSS; { Down Key, Right Key }
1038             puSS : actionSS := staySS;    { Page Up }
1039             pdSS : actionSS := staySS;    { Page Down }
1040                                     { Function Keys }
1041             f1SS, f2SS, f3SS, f4SS, f5SS,
1042             f6SS, f7SS, f8SS, f9SS, f10SS : actionSS := staySS;
1043           End { Case ret };
1044         End
1045       ELSE { retSS is length 1 }
1046         Begin
1047           IF retSS = escSS THEN actionSS := staySS { Escape Key }
1048           End;
1049           { Any other key not in above list will keep actionSS=nextSS }
1050         End; { retrieveSS }

```


AD-A168 517

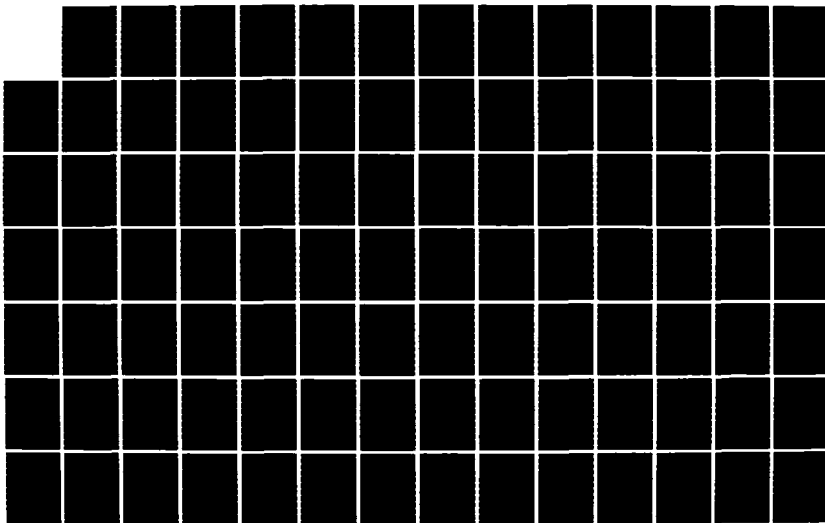
DEVELOPMENT OF AN AUTOMATED MICROCOMPUTER
KNOWLEDGE-BASED INTEGRATED COM. (U) NAVAL POSTGRADUATE
SCHOOL MONTEREY CA R L BEARD MAR 86

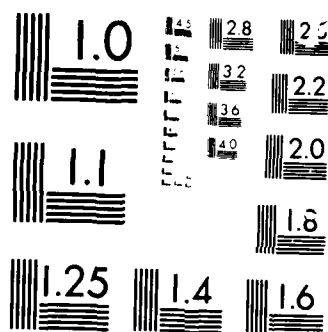
3/5

UNCLASSIFIED

F/G 9/2

NL





Mr. R. H. H.

Mr. R. H. H.

```

1051 CASE actionSS of
1052     staySS: ;
1053     nextSS: Begin
1054         varSS := varSS + 1;
1055         IF varSS > screen_fieldSS THEN varSS := 1;
1056         IF last_fieldSS AND retrieveSS THEN
1057             actionSS := last_field_actionSS
1058     End;
1059     prevSS: Begin
1060         varSS := varSS - 1;
1061         IF varSS < 1 THEN varSS := screen_fieldSS
1062     End;
1063     exitSS: ;
1064 End; { CASE }
1065 End; {PROCEDURE RET_STATUS}
1066
1067
1068 PROCEDURE GETREAL(COL,LIN,LEN :      BYTE;      { Column, Line, Length }
1069                 ITYPE :            CHAR;      { Type= C, N, D, Y, M }
1070                 Var WITEM :        REAL;      { Numerci Variable Name }
1071                 PICT :            STR80;      { Picture X, U, L, 9, 8 # }
1072                 ITEM_LOW,ITEM_HIGH : REAL;      { Range - Numerics/Date Only}
1073                 Var RET :          STR2;      { Returned Code }
1074                 RETRIEVE :         BOOLEAN;   { False=Disp Only, True=Get }
1075                 FGR_COLOR,BGR_COLOR : BYTE);  { Colors Foregr, Backgr }
1076
1077 { This Procedure converts numeric to string before calling GETITEM }
1078 { it then converts the result back to numeric }
1079
1080 Var
1081     numSS, numloSS, numhiSS: STR80;
1082     errorcodeSS,dec_posSS: INTEGER;
1083
1084 Begin
1085     { Get # of Decimal Positions }
1086     dec_posSS:=ord(pict[0])-pos('.',pict);
1087     { Convert item, low and high range to string }
1088     STR (witem:0:dec_posSS,numSS);
1089     STR (item_low:0:dec_posSS,numloSS);
1090     STR (item_high:0:dec_posSS,numhiSS);
1091     GETITEM (col,lin,len,itype,numSS,pict,numloSS,numhiSS,
1092             ret,retrieve,fgr_color,bgr_color);
1093     { Convert string to numeric item }
1094     VAL (numSS, witem, errorcodeSS);
1095 End; { Procedure GETREAL }
1096
1097
1098 PROCEDURE GETINT(COL,LIN,LEN :      BYTE;      { Column, Line, Length }
1099                 ITYPE :            CHAR;      { Type= C, N, D, Y, M }
1100                 Var WITEM :        INTEGER;   { Numerci Variable Name }

```

Page 23

SPLICE.PAS Program Listing

```

1101          PICT :          STR80;    { Picture X, U, L, 9, 8 # }
1102          ITEM_LOW,ITEM_HIGH : INTEGER; { Range - Numerics/Date Only}
1103      Var RET :          STR2;    { Returned Code      }
1104          RETRIEVE :      BOOLEAN; { False=Disp Only, True=Get }
1105          FGR_COLOR,BGR_COLOR : BYTE); { Colors Foregr, Backgr }
1106
1107      { This Procedure converts numeric to string before calling GETITEM }
1108      { It then converts the result back to numeric }
1109
1110      Var
1111          numSS, numloSS, numhiSS: STR80;
1112          errorcodeSS : INTEGER;
1113
1114      Begin
1115          { Convert item, low and high range to string }
1116          STR (witem,numSS);
1117          STR (item_low,numloSS);
1118          STR (item_high,numhiSS);
1119          GETITEM (col,lin,len,itype,numSS,pict,numloSS,numhiSS,
1120              ret,retrieve,fgr_color,bgr_color);
1121          { Convert string to numeric item }
1122          VAL (numSS, witem, errorcodeSS);
1123      End; { Procedure GETINT }
1124
1125
1126
1127      { End of SCREEN SCULPTOR Global Procedures }
1128
1129
1130
1131      PROCEDURE LINE_SETUP;
1132
1133      Var
1134          Temp1 : String [2];
1135          Temp2 : String [4];
1136
1137
1138      Begin { PROCEDURE LINE_SETUP }
1139          IF Siteinfo.siteno < 10 THEN
1140              STR (Siteinfo.siteno:1, Temp1)
1141          ELSE
1142              STR (Siteinfo.siteno:2, Temp1);
1143          Temp2 := Copy (Costtable [1].clin, 1, 4);
1144          IF Siteinfo.siteno < 10 THEN
1145              Line_Number := CONCAT ('0', Temp1, Temp2)
1146          ELSE
1147              Line_Number := CONCAT (Temp1, Temp2);
1148          {*****}
1149          { Accumulate the three totals for each section }
1150          {*****}

```



```

1201
1202
1203 PROCEDURE WRITE_A_LINE;
1204
1205 (*****)
1206 { This procedure is called by two disk file print routines, }
1207 { PRINT_MAINT and PRINT_DOC_or_TRNG to write the data elements }
1208 { associated with each CLIN to the output disk file. }
1209 (*****)
1210
1211 Begin ( Procedure WRITE_A_LINE )
1212     LINE_SETUP;
1213     WRITELN (Diskfile, '', Line_Number:7, ' ', Costtable[I].featureno:8,
1214             ' ', Costtable[I].descript:28, ' ', Quantity:3,
1215             Costtable[I].purchprice:13:2, Extended_Price:12:2,
1216             Costtable[I].basemaint:9:2, Maint_Factor:8:3, ' ',
1217             Costtable[I].basemaint * Maint_Factor * Quantity:12:2,
1218             ' ', ' ', ' ', ' ', ' ', ' ');
1219 End; ( Procedure WRITE_A_LINE )
1220
1221
1222 PROCEDURE PRINT_DOC_or_TRNG;
1223
1224 (*****)
1225 { Sets Parameters for FDC Training Courses and Documentation. }
1226 { Sets both Maint_Months and Maint_Factor to zero (0) }
1227 (*****)
1228
1229 Begin ( Procedure PRINT_DOC_or_TRNG )
1230     Maint_Months := 0; ( No maintenance on training/documentation )
1231     Maint_Factor := 0; ( No maintenance uplift on training/documentation )
1232     Extended_Price := Quantity * Costtable[I].purchprice;
1233     WRITE_A_LINE;
1234 End; ( Procedure PRINT_DOC_or_TRNG )
1235
1236
1237 PROCEDURE COMPUTE_SECTION_TOTALS (Section_Title : Title);
1238
1239 (*****)
1240 { This procedure prints the totals accumulated for each section after the }
1241 { last contract line number and associated data elements are printed. It }
1242 { then prints the title for the next section and prints a new set of }
1243 { headers. After the last contract line number and associated data }
1244 { elements have been printed, the O&MN and OPN totals for each section }
1245 { and the O&MN and OPN grand totals are printed. }
1246 (*****)
1247
1248 Var
1249     E : Integer;
1250     O&MN_Total, OPN_Total, Maint_Totals : Real;

```


Page 27

SPLICE.PAS Program Listing

```

1301      WRITELN (Diskfile, ' ' ' ', 'SUBTOTALS:', ' ' ' ' O&MN',
1302                ' ' ' ', ' ' ' ' OPN');
1303      WRITELN (Diskfile);
1304      WRITELN (Diskfile, ' ' ' ', 'HARDWARE', Totals [0, 1],
1305                ' ' ' ', Totals [0, 2]);
1306      WRITELN (Diskfile, ' ' ' ', 'SOFTWARE', Totals [1, 1],
1307                ' ' ' ', Totals [1, 2]);
1308      WRITELN (Diskfile, ' ' ' ', 'DOCUMENTATION', Totals [2, 1],
1309                ' ' ' ', Totals [2, 2]);
1310      WRITELN (Diskfile, ' ' ' ', 'TRAINING',
1311                Totals [3, 1], ' ' ' ', Totals [3, 2]);
1312      WRITELN (Diskfile, ' ' ' ', 'MAINTENANCE',
1313                Totals [4, 1], ' ' ' ', Totals [4, 2]);
1314      WRITELN (Diskfile, ' ' ' ', 'OTHER', Totals [5, 1],
1315                ' ' ' ', Totals [5, 2]);
1316      WRITELN (Diskfile);
1317      { Initialize O&MN and OPN totals }
1318      OMN_TOTAL := 0;
1319      OPN_TOTAL := 0;
1320      { Compute the O&MN and OPN grand totals. }
1321      FOR K := 0 to 5 DO
1322      Begin
1323          OMN_TOTAL := OMN_TOTAL + Totals [K, 1];
1324          OPN_TOTAL := OPN_TOTAL + Totals [K, 2];
1325      End;
1326      WRITELN (Diskfile, ' ' ' ', 'TOTALS:',
1327                OMN_Total, ' ' ' ', OPN_Total);
1328      End
1329  ELSE
1330      Begin
1331          {*****}
1332          { Print the totals for the section just finished, then print }
1333          { the next section title and new headers. }
1334          {*****}
1335          WRITELN (Diskfile);
1336          WRITELN (Diskfile, ' ' ' ', ' ' ' ', ' ' ' ', 'Totals:',
1337                    Subtotals [ORD (mode), 1], ' ' ' ', ' ' ' ',
1338                    Subtotals [ORD (mode), 2], ' ' ' ',
1339                    Subtotals [ORD (mode), 3]);
1340          WRITELN (Diskfile);
1341          WRITELN (Diskfile, ' ' ' ', Section_Title, ' ');
1342          HEADERS;
1343      End;
1344  End; { Procedure COMPUTE_SECTION_TOTALS }
1345
1346
1347
1348  {-----}
1349
1350      END OF GLOBAL PROCEDURES

```



```

1351 -_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*-*)
1352
1353 PROCEDURE INITIALIZE;
1354
1355
1356
1357 PROCEDURE INIT_TOTALS;
1358
1359 Var
1360     Row, Col : Integer;
1361
1362
1363 {*****}
1364 {   Initialize the subtotals and totals for each section to zero.   }
1365 {*****}
1366 Begin
1367     FOR Row := 0 to 5 DO
1368         FOR Col := 1 to 3 DO
1369             Begin
1370                 Subtotals [Row, Col] := 0;
1371                 IF Col < 3 THEN Totals [Row, Col] := 0;
1372             End;
1373 {*****}
1374 {       Initialize the following global components       }
1375 {*****}
1376 I := 1;                                { Global index counter }
1377 System_Downtime_Component := 0;
1378 Mode := Hard;
1379 BeepOnSS := False;                     { Set to TRUE if sound is desired }
1380 vtypeSS := SET_MONITOR_TYPE;           { 2 = Color, 3 = Monochrome }
1381 TextBackground(1);                     { Initialize background color to BLUE }
1382 ClrScr;                                { Clear the input screen }
1383 ASSIGN (Screenfile, File3);
1384 {$I-}                                  { User responsible for I/O error check }
1385 RESET (Screenfile);
1386 {$I+}                                  { System will check for I/O errors }
1387 End;  { Procedure INIT_TOTALS }
1388
1389
1390 PROCEDURE OPENING_SCREEN;
1391
1392 {*****}
1393 {   This procedure displays the opening screen to the user.   }
1394 {*****}
1395
1396 Begin { Procedure OPENING_SCREEN }
1397     DISPLAY_SCREEN (Screenfile);        { Display Screen }
1398     DELAY (3500);
1399 End;  { Procedure OPENING_SCREEN }
1400

```

```

1401
1402 PROCEDURE PICK_A_SITE;
1403
1404 {*****}
1405 { This procedure has four main functions. First, it determines the site to }
1406 { be configured. Then it obtains the effective date for the delivery order. }
1407 { It THEN obtains the file name for the output file from this session. And }
1408 { finally, it builds the SITE.INFO array which contains site specific data }
1409 { from the CONFIG.SIT file. }
1410 {*****}
1411
1412 Var
1413   Datain           : String [80];
1414   Sitoen, Element  : Integer;
1415   Err, Temp_Site   : Integer;
1416   Textin           : Text;
1417
1418
1419 PROCEDURE GET_SITE_NUMBER;
1420
1421 Begin { Procedure GET_SITE_NUMBER }
1422   { Initialize Variables To Default Values }
1423   Sitoen := 1;
1424
1425   {*****}
1426   { Present the user with a list of the SPLICE sites by name and number. }
1427   {*****}
1428
1429   screen_fieldSS := 1;
1430   varSS := 1;
1431   retrieveSS := FALSE;
1432   last_fieldSS := FALSE;
1433   DISPLAY_SCREEN (Screenfile);      { Display Screen }
1434
1435   REPEAT { until answerSS = 'Y' }
1436   { Display Items. Change retrieveSS to TRUE and INPUT items }
1437   REPEAT { until actionSS = exitSS }
1438     REPEAT
1439       GETINT(69,24,2,'N',Sitoen,'##',1,58,retSS,retrieveSS,14,1);
1440       IF Sitoen = 23 THEN
1441         Begin
1442           GoToXY (20, 25);
1443           Color (15, 4);
1444           WRITE ('G,' Site INACTIVE and not available for selection ');
1445         End;
1446       UNTIL Sitoen > 23;
1447       IF varSS = screen_fieldSS THEN last_fieldSS := TRUE;
1448       RET_STATUS; { Check the code in "retSS". Set "varSS" and "actionSS" }
1449
1450       { Check to see whether to switch retrieveSS to true }

```

```

1451     IF last_fieldSS and (not retrieveSS) THEN
1452         Begin
1453             retrieveSS := TRUE;
1454             last_fieldSS := FALSE;
1455             actionSS := staySS;
1456             varSS := 1;
1457         End
1458     ELSE
1459         last_fieldSS := FALSE;
1460     UNTIL actionSS = exitSS;
1461     ACCEPT_INPUTS;
1462     UNTIL answerSS = 'Y';
1463 End; { Procedure GET_SITE_NUMBER }
1464
1465
1466 Begin { Procedure PICK_A_SITE }
1467     GET_SITE_NUMBER;
1468     ASSIGN (Textin, File2);
1469     RESET (Textin);
1470     { Initialize "Temp_Site" and "Stock_Point" }
1471     Temp_Site := 0;
1472     Stock_Point := ' ';
1473     WHILE Not EOF (Textin) AND (Temp_Site < Sitenos) DO
1474     { *****
1475     {   Read the file "CONFIG.SIT" until the site number in the file is
1476     {   equal to the site number input by the user.
1477     { *****
1478     Begin
1479         READLN (Textin, Datain);
1480         Val (Copy (Datain, 1, 2), Temp_Site, Err);
1481
1482         { Is site # from COSTS.IN = site # selected for configuration? }
1483         IF Sitenos = Temp_Site THEN
1484             Begin
1485                 { Builds the site information record }
1486                 SiteInfo.sitenos := sitenos;
1487                 SiteInfo.sitename := Copy (Datain, 3, 27);
1488                 Val (Copy (Datain, 31, 1), SiteInfo.documentation, Err);
1489                 Val (Copy (Datain, 33, 1), SiteInfo.training, Err);
1490                 SiteInfo.maint_options := Copy (Datain, 35, 4);
1491                 SiteInfo.maint_response := Copy (Datain, 40, 1);
1492                 SiteInfo.site_type := Copy (Datain, 42, 1);
1493                 Val (Copy (Datain, 44, 6), SiteInfo.site_inst_cost, Err);
1494             End;
1495         End;
1496         Stock_Point := SiteInfo.site_type;
1497         CLOSE (Textin);
1498     End; { Procedure PICK_A_SITE }
1499
1500

```

```

1501 PROCEDURE BUILD_COST_TABLE;
1502
1503 {*****}
1504 { This procedure's primary function is to build the COSTTABLE array. This }
1505 { contains the identification data for each component from the COSTS.IN file }
1506 { as well as cost/maintenance data, which is updated by the applicable up- }
1507 { lift or discount factors. The array currently contains room for 200 }
1508 { entries. }
1509 {*****}
1510
1511 Var
1512   Textin : Text;
1513   Datain : String [80];           ( Data coming in from COSTS.IN file )
1514   Errorcode, Count : Integer;
1515   LCN_Purch_Esc_Rate, LCN_Momaint_Esc_Rate, Document_Esc_Rate : Real;
1516   Purch_Esc_Rate, Instal_Esc_Rate, Train_Esc_Rate : Real;
1517   SPLICENet_SW_Maint_Esc_Rate, SPLICENet_SW_Purch_Esc_Rate : Real;
1518   FDC_SNA_Purch_Esc_Rate, LCN_SW_Esc_Rate : Real;
1519
1520 PROCEDURE GET_RATES;
1521 {*****}
1522 { This procedure serves three main functions: it obtains the name of the }
1523 { current user, then obtains all the escalation/discount rates, and finally }
1524 { several numbers of Maint_Months, which are used for maintenance calculations. }
1525 {*****}
1526
1527 Var
1528   Month_Index : String [2];
1529   FRN_Name, Effective_Date : String [8];
1530   Index, Position : Integer;
1531
1532
1533 PROCEDURE INITIALIZE_RATES;
1534
1535 {Initialize Variables To Default Values}
1536
1537 Begin ( Procedure INITIALIZE_RATES )
1538
1539   Purch_Esc_Rate := 0.00;
1540   LCN_Purch_Esc_Rate := 0.00;
1541   SPLICENet_SW_Maint_Esc_Rate := 0.00;
1542   SPLICENet_SW_Purch_Esc_Rate := 0.00;
1543   Emerg_Maint_Rate := 0.0;
1544   FDC_SNA_Purch_Esc_Rate := 0.00;
1545   LCN_Momaint_Esc_Rate := 0.000;
1546   LCN_SW_Esc_Rate := 0.000;
1547   Instal_Esc_Rate := 0.000;
1548   Train_Esc_Rate := 0.00;
1549   Document_Esc_Rate := 0.00;
1550   Momaint_Esc_Rate := 0.000;

```

```

1601      10: GETREAL(71,17,4,'N',Train_Esc_Rate,
1602          '###',0.00,9.99,retSS,retrieveSS,15,3);
1603      11: GETREAL(70,18,5,'N',Document_Esc_Rate,
1604          '###',-1.00,9.99,retSS,retrieveSS,15,3);
1605      12: GETREAL(70,19,5,'N',MomaInt_Esc_Rate,
1606          '###',0.000,9.999,retSS,retrieveSS,15,3);
1607      13: GETITEM(63,21,8,'C',PRN_NAME,
1608          'UUUUUUU',' ',retSS,retrieveSS,15,3);
1609      14: GETINT(37,23,2,'N',Maint_Months,
1610          '##',0,12,retSS,retrieveSS,15,3);
1611      15: GETITEM(67,23,8,'D',Effective_Date,
1612          '88/88/88','01/01/84','12/31/99',retSS,retrieveSS,15,3);
1613      End;      { CASE }
1614
1615      IF varSS = screen_fieldSS THEN last_fieldSS := TRUE;
1616      RET_STATUS;      { Check code in "retSS". Set "varSS" & "actionSS" }
1617
1618      { Check to see whether to switch retrieveSS to true }
1619      IF last_fieldSS AND (not retrieveSS) THEN
1620          Begin
1621              retrieveSS := TRUE;
1622              last_fieldSS := FALSE;
1623              actionSS := staySS;
1624              varSS := 1;
1625          End
1626      ELSE
1627          last_fieldSS := FALSE;
1628      UNTIL actionSS = exitSS;
1629      ACCEPT_INPUTS;
1630      UNTIL answerSS = 'Y';
1631      End;      { Procedure GET_RATE_INPUTS }
1632
1633
1634      Begin { Procedure GET_RATES }
1635          INITIALIZE_RATES;
1636          GET_RATE_INPUTS;
1637          { Generate the correct escalation & discount rates }
1638          FDC_SNA_Purch_Esc_Rate := FDC_SNA_Purch_Esc_Rate + 1;
1639          Purch_Esc_Rate := 1 - Purch_Esc_Rate;
1640          LCN_Purch_Esc_Rate := 1 - LCN_Purch_Esc_Rate;
1641          SPLICENet_SW_Maint_Esc_Rate := SPLICENet_SW_Maint_Esc_Rate + 1;
1642          SPLICENet_SW_Purch_Esc_Rate := SPLICENet_SW_Purch_Esc_Rate + 1;
1643          Instal_Esc_Rate := 1 + Instal_Esc_Rate;
1644          Document_Esc_Rate := 1 + Document_Esc_Rate;
1645          MomaInt_Esc_Rate := MomaInt_Esc_Rate + 1;
1646          Train_Esc_Rate := 1 + Train_Esc_Rate;
1647          LCN_MomaInt_Esc_Rate := 1 + LCN_MomaInt_Esc_Rate;
1648          LCN_SW_Esc_Rate := 1 + LCN_SW_Esc_Rate;
1649          Emerg_Maint_Rate := 1 + Emerg_Maint_Rate;
1650          { Generate the complete output file name, with LOGUS 1, 2, 3 "P" extension }

```

Page 32

SPLICE.PAS Program Listing

```

1551 PRN_Name := 'SPLICE';
1552 Maint_Months := 0;
1553 Effective_Date := '09/01/85';
1554 End; { Procedure INITIALIZE_RATES }
1555
1556
1557 PROCEDURE GET_RATE_INPUTS;
1558
1559 Begin { Procedure GET_RATE_INPUTS }
1560   screen_fieldSS := 15;
1561   varSS := 1;
1562   retrieveSS := FALSE;
1563   last_fieldSS := FALSE;
1564   DISPLAY_SCREEN (Screenfile); { Display Screen }
1565   { If the site selected is a MAP site, blank out the fields related to
1566     HYPERchannel (ICN) escalation and discount rates.
1567   IF Stock_Point <> 'S' THEN
1568     Begin
1569       COLOR (1, 1);
1570       GOTOXY (70, 11);
1571       WRITE (' ');
1572       GOTOXY (70, 15);
1573       WRITE (' ');
1574     End;
1575
1576   REPEAT { until answerSS = 'Y' }
1577   { Display Items. Change retrieveSS to TRUE and INPUT items}
1578   REPEAT { until actionSS = exitSS }
1579     CASE varSS of
1580       1: GETREAL(71,8,4,'N',FDC_SWA_Purch_Esc_Rate,
1581         '#.##',0.00,9.99,retSS,retrieveSS,15,3);
1582       2: GETREAL(71,9,4,'N',Purch_Esc_Rate,
1583         '#.##',0.00,9.99,retSS,retrieveSS,15,3);
1584       3: IF Stock_point = 'S' THEN
1585         GETREAL(71,10,4,'N',ICN_Purch_Esc_Rate,
1586           '#.##',0.00,9.99,retSS,retrieveSS,15,3);
1587       4: GETREAL(71,11,4,'N',SPLICNet_SW_Maint_Esc_Rate,
1588         '#.##',0.00,9.99,retSS,retrieveSS,15,3);
1589       5: GETREAL(71,12,4,'N',SPLICNet_SW_Purch_Esc_Rate,
1590         '#.##',0.00,9.99,retSS,retrieveSS,15,3);
1591       6: GETREAL(72,13,3,'N',Emerg_Maint_Rate,
1592         '#.##',0.0,9.9,retSS,retrieveSS,15,3);
1593       7: IF Stock_Point = 'S' THEN
1594         GETREAL(70,14,5,'N',ICN_Maint_Esc_Rate,
1595           '#.###',0.000,9.999,retSS,retrieveSS,15,3);
1596       8: IF Stock_Point = 'S' THEN
1597         GETREAL(70,15,5,'N',ICN_SW_Esc_Rate,
1598           '#.###',0.000,9.999,retSS,retrieveSS,15,3);
1599       9: GETREAL(70,16,5,'N',Instal_Esc_Rate,
1600         '#.###',0.000,9.999,retSS,retrieveSS,15,3);

```

```

1651 PRN_File_Name := CONCAT (PRN_Name, '.PRN');
1652 Day := Copy (Effective_Date, 4, 2);
1653 Month_Index := Copy (Effective_Date, 1, 2);
1654 Val (Month_Index, Index, Errorcode);
1655 Month := Month_Name [Index];
1656 { Strip trailing blanks off the name of the month }
1657 Position := POS (' ', Month);
1658 IF Position <> 0 THEN Month := Copy (Month, 1, Position - 1);
1659 Year := Copy (Effective_Date, 7, 2);
1660 Year := CONCAT ('19', Year);
1661 End; { Procedure GET_RATES }
1662
1663
1664 Begin { Procedure BUILD_COST_TABLE }
1665   ASSIGN (Textin, File1);
1666   RESET (Textin);
1667   Count := 1;
1668   GET_RATES; {ask user for all discount and escalation rates to be used }
1669   ClrScr;
1670   COLOR (15, 1);
1671   GOTOXY (16, 13);
1672   WRITE ('Constructing cost escalation and discount table. ');
1673   READLN (Textin, Datain);
1674   WHILE Not EOF (TEXTIN) DO
1675     Begin
1676       { Build the Costtable array }
1677       Costtable [Count].featureno := Copy (Datain, 6, 6);
1678       Costtable [Count].clin := Copy (Datain, 1, 4);
1679       Costtable [Count].descript := Copy (Datain, 13, 27);
1680       Val (Copy (Datain, 40, 10), Costtable [Count].momaint, Errorcode);
1681       Costtable [Count].basemaint := Costtable [Count].momaint;
1682
1683       { LCN H/W Base Maintenance }
1684       IF (Costtable [Count].featureno > '320100') AND
1685         (Costtable [Count].featureno < '420400') THEN
1686         Costtable [Count].basemaint := Costtable [Count].momaint
1687           * LCN_Momaint_Esc_Rate;
1688
1689       { LCN S/W Base Maintenance }
1690       ELSE IF (Costtable [Count].featureno = '550801') OR
1691         (Costtable [Count].featureno = '550901') OR
1692         (Costtable [Count].featureno = '551001') OR
1693         (Costtable [Count].featureno = '551101') OR
1694         (Costtable [Count].featureno = '551201') OR
1695         (Costtable [Count].featureno = '551301') THEN
1696         Costtable [Count].basemaint := Costtable [Count].momaint
1697           * LCN_SW_Esc_Rate;
1698
1699       { SPLICENet S/W Base Maintenance }
1700       ELSE IF (Costtable [Count].featureno = '550710') OR

```

```

1701      (Costtable [Count].featureno = '550711') OR
1702      (Costtable [Count].featureno = '550803') OR
1703      (Costtable [Count].featureno = '550903') OR
1704      (Costtable [Count].featureno = '551003') OR
1705      (Costtable [Count].featureno = '551103') OR
1706      (Costtable [Count].featureno = '551203') OR
1707      (Costtable [Count].featureno = '551303') OR
1708      (Costtable [Count].featureno = '551304') OR
1709      (Costtable [Count].featureno = '551403') OR
1710      (Costtable [Count].featureno = '551500') OR
1711      (Costtable [Count].featureno = '551501') OR
1712      (Costtable [Count].featureno = '551502') OR
1713      (Costtable [Count].featureno = '551503') OR
1714      (Costtable [Count].featureno = '551504') THEN
1715      Costtable [Count].basemaint := Costtable [Count].momaint
1716      * SPLICENet_SW_Maint_Esc_Rate
1717
1718      ( Normal Maintenance Escalation )
1719      ELSE Costtable [Count].momaint := Costtable [Count].momaint
1720      * Momaint_Esc_Rate;
1721
1722      ( 6100 H/W Purchase Escalation )
1723      Val (Copy (Datain, 50, 11), Costtable [Count].purchprice, Errorcode);
1724      IF (Costtable [Count].featureno > '450300') AND
1725      (Costtable [Count].featureno < '450400') THEN
1726      Costtable [Count].purchprice := Costtable [Count].purchprice
1727
1728      ( 6100 S/W Purchase Escalation )
1729      ELSE IF (Costtable [Count].featureno > '550701') AND
1730      (Costtable [Count].featureno < '550710') THEN
1731      Costtable [Count].purchprice := Costtable [Count].purchprice
1732
1733      (SPLICENet S/W Base Maintenance)
1734      ELSE IF (Costtable [Count].featureno = '550710') OR
1735      (Costtable [Count].featureno = '550711') OR
1736      (Costtable [Count].featureno = '550803') OR
1737      (Costtable [Count].featureno = '550903') OR
1738      (Costtable [Count].featureno = '551003') OR
1739      (Costtable [Count].featureno = '551103') OR
1740      (Costtable [Count].featureno = '551203') OR
1741      (Costtable [Count].featureno = '551303') OR
1742      (Costtable [Count].featureno = '551304') OR
1743      (Costtable [Count].featureno = '551403') OR
1744      (Costtable [Count].featureno = '551500') OR
1745      (Costtable [Count].featureno = '551501') OR
1746      (Costtable [Count].featureno = '551502') OR
1747      (Costtable [Count].featureno = '551503') OR
1748      (Costtable [Count].featureno = '551504') THEN
1749      Costtable [Count].basemaint := Costtable [Count].momaint
1750      * SPLICENet_SW_Purch_Esc_Rate

```



```

1751
1752 { Training Escalation }
1753 ELSE IF (Costtable [Count].featureno = '39XXXX') or
1754         (Costtable [Count].featureno = 'XXXXXX') THEN
1755     Costtable [Count].purchprice := Costtable [Count].purchprice
1756                                   * Train_Esc_Rate
1757
1758 { LCN H/W Purchase Escalation }
1759 ELSE IF (Costtable [Count].featureno > '320100') AND
1760         (Costtable [Count].featureno < '420400') THEN
1761     Costtable [Count].purchprice := Costtable [Count].purchprice
1762                                   * LCN_Purch_Esc_Rate
1763
1764 { FDC SNA Purchase Escalation }
1765 ELSE IF (Costtable [Count].featureno = '550710') THEN
1766     Costtable [Count].purchprice := Costtable [Count].purchprice
1767                                   * FDC_SNA_Purch_Esc_Rate
1768
1769 { LCN S/W Purchase Escalation }
1770 ELSE IF (Costtable [Count].featureno = '550801') OR
1771         (Costtable [Count].featureno = '550901') OR
1772         (Costtable [Count].featureno = '551001') OR
1773         (Costtable [Count].featureno = '551101') OR
1774         (Costtable [Count].featureno = '551201') OR
1775         (Costtable [Count].featureno = '551301') THEN
1776     Costtable [Count].purchprice := Costtable [Count].purchprice
1777                                   * LCN_SW_Esc_Rate
1778
1779 { Documentation Purchase Escalation }
1780 ELSE IF (Costtable [Count].featureno > '710000') AND
1781         (Costtable [Count].featureno < '749999') THEN
1782     Costtable [Count].purchprice := Costtable [Count].purchprice
1783                                   * Document_Esc_Rate
1784
1785 { Site Preparation Installation Escalation }
1786 ELSE IF Costtable [Count].featureno = '000101' THEN
1787     Costtable [Count].purchprice := SiteInfo.site_inst_cost
1788                                   * Instal_Esc_Rate
1789
1790 { Normal S/W Purchase Escalation }
1791 ELSE Costtable [Count].purchprice := Costtable [Count].purchprice
1792                                   * Purch_Esc_Rate;
1793
1794 { Installation Cost Escalation }
1795 Val (Copy (Datain, 62, 10), Costtable [Count].instcost, Errorcode);
1796 IF (Costtable [Count].featureno > '450300') AND
1797     (Costtable [Count].featureno < '450400') THEN
1798     Costtable [Count].instcost := Costtable [Count].instcost
1799 ELSE IF (Costtable [Count].featureno > '550701') AND
1800     (Costtable [Count].featureno < '550800') THEN

```

(Name of work processing method) : _____

Page 38

SPLICE.PAS-include file SPLICE2.PAS Program Listing

```

1851I  { Input Variables Used For Documentation, Training & Maintenance }
1852I  Computer_Ops, Hardware_Manual, Programmer_Ref      : Integer;
1853I  Sys_Programmer, Training_Group, Data_Communication : Integer;
1854I  Hardware_Overview, Operator_Training, Sys_Resource : Integer;
1855I  SPLICNet_Workshop, Sys_Tuning_Xray, TAL, Per_Call_Months : Integer;
1856I
1857I
1858I  PROCEDURE CONFIGURE_HARDWARE;
1859I
1860I  Var
1861I    Cable_Distance : String [1];
1862I    Add_Expansion, Add_HYPERchannel, Add_Patchpanel, Add_System      : Integer;
1863I    AsyncCtrl, AsyncExtbdl, AsyncPchpnl, A510, BitSync, ByteSync    : Integer;
1864I    Crts, D128MB, D240MB, D540MB, ExpanCab                        : Integer;
1865I    HYPERCab, LPM1000, LPM600, PatchPanel, Printers, RdrPunch       : Integer;
1866I    SysCab, TapeDrv, Trunks                                         : Integer;
1867I
1868I
1869I  PROCEDURE INITIALIZE_HARDWARE_INPUTS;
1870I
1871I  Begin { Procedure INITIALIZE_HARDWARE_INPUTS }
1872I    { Initialize Variables To Default Values }
1873I    Add_Expansion := 0;
1874I    Add_HYPERchannel := 0;
1875I    Add_Patchpanel := 0;
1876I    Add_System := 0;
1877I    AsyncCtrl := 0;
1878I    AsyncExtbdl := 0;
1879I    AXXX := 0;
1880I    A140 := 0;
1881I    A150 := 0;
1882I    A220 := 0;
1883I    A400 := 0;
1884I    A510 := 0;
1885I    BitSync := 0;
1886I    ByteSync := 0;
1887I    Cable_Distance := 'B';
1888I    CarRdr := 0;
1889I    Crts := 0;
1890I    D128MB := 0;
1891I    D240MB := 0;
1892I    D540MB := 0;
1893I    HYPERcab := 0;
1894I    LIU := 0;
1895I    LPM1000 := 0;
1896I    LPM600 := 0;
1897I    Processors := 0;
1898I    Printers := 0;
1899I    RdrPunch := 0;
1900I    TapeDrv := 0;

```

```

19011    THYPERchannels := 0;
19021    Trunks := 0;
19031 End;    { Procedure INITIALIZE_HARDWARE_INPUTS }
19041
19051 PROCEDURE ODD_ERROR;
19061
19071 Begin { Procedure ODD_ERROR }
19081    CLAR (15, 4);
19091    GOTOXY (18, 25);
19101    WRITE ('G, ' Number of disks must be 0 or an EVEN number! ');
19111 End;    { Procedure ODD_ERROR }
19121
19131
19141 PROCEDURE CLEAR_MESSAGE;
19151
19161 Begin { Procedure CLEAR_MESSAGE }
19171    TextBackground (1);
19181    GOTOXY (1, 25);
19191    ClrEol;
19201 End;    { Procedure CLEAR_MESSAGE }
19211
19221
19231 PROCEDURE GET_HARDWARE_INPUTS;
19241
19251 Begin { Procedure GET_HARDWARE_INPUTS }
19261    screen_fieldSS := 25;
19271    varSS := 1;
19281    retrieveSS := False;
19291    last_fieldSS := False;
19301    DISPLAY SCREEN (Screenfile);    { Display Screen }
19311
19321    REPEAT (until answerSS = 'Y' )
19331    { Display Items. Change retrieveSS to True and INPUT items}
19341    REPEAT ( until actionSS = exitSS )
19351        CASE varSS of
19361            1: GETINT(40,4,3,'N',Processors,'###',0,256,retSS,retrieveSS,14,1);
19371            2: GETINT(40,5,3,'N',Printers,'###',0,12,retSS,retrieveSS,14,1);
19381            3: GETINT(40,6,3,'N',Crts,'###',0,999,retSS,retrieveSS,14,1);
19391            4: REPEAT
19401                GETINT(40,7,3,'N',D128MB,'###',0,128,retSS,retrieveSS,14,1);
19411                IF ODD (D128MB) THEN ODD_ERROR
19421                ELSE CLEAR_MESSAGE;
19431            UNTIL not ODD (D128MB);
19441            5: REPEAT
19451                GETINT(40,8,3,'N',D240MB,'###',0,128,retSS,retrieveSS,14,1);
19461                IF ODD (D240MB) THEN ODD_ERROR
19471                ELSE CLEAR_MESSAGE;
19481            UNTIL not ODD (D240MB);
19491            6: REPEAT
19501                GETINT(40,9,3,'N',D540MB,'###',0,128,retSS,retrieveSS,14,1);

```

```

1951I      IF ODD (D540MB) THEN ODD_ERROR
1952I      ELSE CLEAR MESSAGE;
1953I      UNTIL not ODD (D540MB);
1954I      7: GETINT(40,10,3,'N',AsyncCtrl,'###',0,64,retSS,retrieveSS,14,1);
1955I      8: GETINT(40,11,3,'N',AsyncExtbd,'###',0,2,retSS,retrieveSS,14,1);
1956I      9: GETINT(40,12,3,'N',BitSync,'###',0,128,retSS,retrieveSS,14,1);
1957I      10: GETINT(40,13,3,'N',ByteSync,'###',0,128,retSS,retrieveSS,14,1);
1958I      11: GETINT(40,14,3,'N',TapeDrv,'###',0,128,retSS,retrieveSS,14,1);
1959I      12: GETINT(40,15,3,'N',RdrPunch,'###',0,12,retSS,retrieveSS,14,1);
1960I      13: GETINT(40,16,3,'N',CardRdr,'###',0,12,retSS,retrieveSS,14,1);
1961I      14: GETINT(40,17,3,'N',LPM1000,'###',0,16,retSS,retrieveSS,14,1);
1962I      15: GETINT(40,18,3,'N',LPM600,'###',0,16,retSS,retrieveSS,14,1);
1963I      16: IF Stock_Point = 'S' THEN
1964I          GETINT(40,19,3,'N',Trunks,'###',0,2,retSS,retrieveSS,14,1);
1965I      17: GETINT(40,20,3,'N',LIU,'###',0,256,retSS,retrieveSS,14,1);
1966I      18: IF Stock_Point = 'S' THEN
1967I          GETINT(73,3,3,'N',A400,'###',0,256,retSS,retrieveSS,14,1);
1968I      19: IF Stock_Point = 'S' THEN
1969I          GETINT(73,4,3,'N',A150,'###',0,256,retSS,retrieveSS,14,1);
1970I      20: IF Stock_Point = 'S' THEN
1971I          GETINT(73,5,3,'N',AXXX,'###',0,256,retSS,retrieveSS,14,1);
1972I      21: IF Stock_Point = 'S' THEN
1973I          GETINT(73,6,3,'N',A220,'###',0,256,retSS,retrieveSS,14,1);
1974I      22: IF Stock_Point = 'S' THEN
1975I          GETINT(73,7,3,'N',A140,'###',0,256,retSS,retrieveSS,14,1);
1976I      23: IF Stock_Point = 'S' THEN
1977I          GETINT(73,8,3,'N',A510,'###',0,256,retSS,retrieveSS,14,1);
1978I      24: IF Stock_Point = 'S' THEN
1979I          GETINT(73,9,3,'N',THYPERchannels,
1980I              '###',0,128,retSS,retrieveSS,14,1);
1981I      25: IF Stock_Point = 'S' THEN
1982I          REPEAT
1983I              GETITEM(75,20,1,'C',Cable_Distance,
1984I                  'U','','',retSS,retrieveSS,14,1);
1985I              IF (Cable_Distance < 'A') OR (Cable_Distance > 'F') THEN
1986I                  Begin
1987I                      COLOR (15, 4);
1988I                      GXYOXY (28, 25);
1989I                      WRITE (^G, ' Not within range A to F ');
1990I                  End
1991I              ELSE CLEAR MESSAGE;
1992I          UNTIL (Cable_Distance >= 'A') AND (Cable_Distance <= 'F');
1993I      End; { CASE }
1994I
1995I      IF varSS = screen_fieldSS THEN last_fieldSS := True;
1996I      RET STATUS; { Check code in "retSS". Set "varSS" and "actions" }
1997I
1998I      { Check to see whether to switch retrieveSS to true }
1999I      IF last_fieldSS AND (not retrieveSS) THEN
2000I          Begin

```

Page 41

SPLICE.PAS-include file SPLICE2.PAS Program Listing

```

2001I         retrieveSS := True;
2002I         last_fieldSS := False;
2003I         actionSS := staySS;
2004I         varSS := 1;
2005I         End
2006I     ELSE
2007I         last_fieldSS := False;
2008I     UNTIL actionSS = exitSS;
2009I     ACCEPT INPUTS;
2010I     UNTIL answerSS = 'Y';
2011I End; { Procedure GET_HARDWARE_INPUTS }
2012I
2013I
2014I PROCEDURE ADDITIONAL_CABINETS;
2015I
2016I Begin { Procedure ADDITIONAL_CABINETS }
2017I     screen_fieldSS := 3;
2018I     varSS := 1;
2019I     retrieveSS := False;
2020I     last_fieldSS := False;
2021I     DISPLAY_SCREEN (Screenfile); { Display Screen }
2022I
2023I     GETINT(40,4,3,'N',Processors,'###',0,256,retSS,False,14,1);
2024I     GETINT(40,5,3,'N',Printers,'###',0,12,retSS,False,14,1);
2025I     GETINT(40,6,3,'N',Crts,'###',0,999,retSS,False,14,1);
2026I     GETINT(40,7,3,'N',D128MB,'###',0,128,retSS,False,14,1);
2027I     GETINT(40,8,3,'N',D240MB,'###',0,128,retSS,False,14,1);
2028I     GETINT(40,9,3,'N',D540MB,'###',0,128,retSS,False,14,1);
2029I     GETINT(40,10,3,'N',AsyncCtrl,'###',0,64,retSS,False,14,1);
2030I     GETINT(40,11,3,'N',AsyncExtbd,'###',0,2,retSS,False,14,1);
2031I     GETINT(40,12,3,'N',BitSync,'###',0,128,retSS,False,14,1);
2032I     GETINT(40,13,3,'N',ByteSync,'###',0,128,retSS,False,14,1);
2033I     GETINT(40,14,3,'N',TapeDrv,'###',0,128,retSS,False,14,1);
2034I     GETINT(40,15,3,'N',RdrPunch,'###',0,12,retSS,False,14,1);
2035I     GETINT(40,16,3,'N',CardRdr,'###',0,12,retSS,False,14,1);
2036I     GETINT(40,17,3,'N',LPM1000,'###',0,16,retSS,False,14,1);
2037I     GETINT(40,18,3,'N',LPM600,'###',0,16,retSS,False,14,1);
2038I     IF Stock_Point = 'S' THEN
2039I         GETINT(40,19,3,'N',Trunks,'###',0,2,retSS,False,14,1);
2040I     GETINT(40,20,3,'N',LIU,'###',0,256,retSS,False,14,1);
2041I     IF Stock_Point = 'S' THEN
2042I         Begin
2043I             GETINT(73,3,3,'N',A400,'###',0,256,retSS,False,14,1);
2044I             GETINT(73,4,3,'N',A150,'###',0,256,retSS,False,14,1);
2045I             GETINT(73,5,3,'N',AXXX,'###',0,256,retSS,False,14,1);
2046I             GETINT(73,6,3,'N',A220,'###',0,256,retSS,False,14,1);
2047I             GETINT(73,7,3,'N',A140,'###',0,256,retSS,False,14,1);
2048I             GETINT(73,8,3,'N',A510,'###',0,256,retSS,False,14,1);
2049I             GETINT(73,9,3,'N',HYPERchannels,'###',0,128,retSS,False,14,1);
2050I             GETINT(75,20,1,'C',Cable_Distance,'0','','',retSS,False,14,1);

```

```

20511      End;
20521      GETINT(68,13,2,'N',PatchPanel,'##',0,16,retSS,False,14,1);
20531      GETINT(68,14,2,'N',SysCab,'##',0,16,retSS,False,14,1);
20541      GETINT(68,15,2,'N',ExpanCab,'##',0,16,retSS,False,14,1);
20551
20561      REPEAT ( until answerSS = 'Y' )
20571      ( Display Items. Change retrieveSS to True and INPUT items)
20581      REPEAT ( until actionSS = exitSS )
20591          CASE varSS of
20601              1: GETINT(75,13,2,'N',Add_PatchPanel,
20611                  '##',0,8,retSS,retrieveSS,14,1);
20621              2: GETINT(75,14,2,'N',Add_System,
20631                  '##',0,8,retSS,retrieveSS,14,1);
20641              3: GETINT(75,15,2,'N',Add_Expansion,
20651                  '##',0,8,retSS,retrieveSS,14,1);
20661          End; ( CASE )
20671
20681          IF varSS = screen_fieldSS THEN last_fieldSS := True;
20691          RET_STATUS; ( Check the code in "retSS". Set "varSS" and "actionSS" )
20701
20711          ( Check to see whether to switch retrieveSS to true )
20721          IF last_fieldSS AND (not retrieveSS) THEN
20731              Begin
20741                  retrieveSS := True;
20751                  last_fieldSS := False;
20761                  actionSS := staySS;
20771                  varSS := 1;
20781              End
20791          ELSE
20801              last_fieldSS := False;
20811          UNTIL actionSS = exitSS;
20821          ACCEPT_INPUTS;
20831          UNTIL answerSS = 'Y';
20841      End; ( Procedure ADDITIONAL_CABINETS )
20851
20861
20871      PROCEDURE PRINT_HW;
20881      { *****
20891      {   This routine is used in the hardware generation process to set up the
20901      {   necessary parameters to be used by PRINT_HW when called.
20911      { *****
20921
20931      Begin ( PROCEDURE PRINT_HW )
20941          Maint_Factor := Momaint_Esc_Rate;
20951          Extended_Price := Quantity * CostTable[1].purchase;
20961          LINE_SETUP;
20971          { *****
20981          {   Compute System Downtime Credit Component Factor per month
20991          { *****
21001          System_Downtime_Component := System_Downtime_Component +

```

```

2101I          (Quantity * CostTable[I].basemaint
2102I          * Maint_Factor);
2103I          {*****}
2104I          { Compute the Component Downtime Credit Factor per hour }
2105I          {*****}
2106I  Downtime_Credit := (((CostTable[I].purchprice + CostTable[I].instcost) / 48)
2107I          + (CostTable[I].basemaint * Maint_factor)) * 0.005;
2108I  WRITELN (Diskfile, "", Line_Number:7, " ", CostTable[I].featureno:8,
2109I          " ", CostTable[I].descript:28, " ", Quantity:3,
2110I          CostTable[I].purchprice:13:2, Extended_Price:12:2,
2111I          CostTable[I].basemaint:9:2, Maint_Factor:8:3, Maint_Months:5,
2112I          Quantity * CostTable[I].basemaint * Maint_Factor
2113I          * Maint_Months:12:2, CostTable[I].instcost:8:2,
2114I          CostTable[I].instcost * Quantity:9:2, Downtime_Credit:9:2,
2115I          (Quantity * CostTable[I].basemaint * Maint_Factor):9:2);
2116I End; { Procedure PRINT_HW }
2117I
2118I
2119I PROCEDURE CONFIGURE_PROCESSING_SUBSYSTEM;
2120I
2121I Var
2122I   OSP : Integer;
2123I
2124I
2125I PROCEDURE COMPUTE_PROCESSORS;
2126I {*****}
2127I { This procedure outputs a series of screens prompting the user to pro- }
2128I { vide the necessary inputs required to generate the processor related }
2129I { data for the desired configuration. Each input is checked to determine }
2130I { whether OR not the response is positive OR within the necessary limits. }
2131I {*****}
2132I
2133I Begin { Procedure COMPUTE_PROCESSORS }
2134I   Quantity := Processors;
2135I   I := I + 1; { I=2 Processors on delivery order }
2136I   IF Quantity > 0 THEN PRINT_HW;
2137I   I := I + 1; { I=3 Uses # of Processors to determine }
2138I               # extra 2MB memory modules to order }
2139I   IF Quantity > 0 THEN PRINT_HW;
2140I   IF (SiteInfo.sitenno = 2) OR (SiteInfo.sitenno = 3) THEN
2141I     Begin
2142I       I := I + 1; { I=4 Floating Point Arithmetic, }
2143I                   only ordered by EMSO sites }
2144I       IF Quantity > 0 THEN PRINT_HW;
2145I     End
2146I   ELSE I := I + 1;
2147I   { The following routine determines the number of OSPs to order. }
2148I   { One OSP is required per 16 Processors. }
2149I   I := I + 1; { I=5 OSP }
2150I   OSP := Processors;

```



```

2151I  WHILE OSP MOD 16 > 0 DO
2152I  OSP := OSP + 1;
2153I  Quantity := OSP DIV 16;
2154I  IF Quantity > 0 THEN PRINT_HW;
2155I End;  { Procedure COMPUTE_PROCESSORS }
2156I
2157I
2158I PROCEDURE COMPUTE_CRTS_PTRS;
2159I {*****}
2160I { This routine computes the number of Centronics Printers, CRTs and OSP
2161I { interfaces required on the delivery order.
2162I {*****}
2163I
2164I Begin {Procededure COMPUTE_CRTS_PTRS }
2165I   Quantity := Printers;
2166I   I := I + 1;           { I=6 Serial Printers }
2167I   IF Quantity > 0 THEN PRINT_HW;
2168I   Quantity := Crts;
2169I   I := I + 1;           { I=7 Crts }
2170I   IF Quantity > 0 THEN PRINT_HW;
2171I   Quantity := OSP DIV 16;
2172I   I := I + 1;           { I=8 Printer Interfaces for OSPs }
2173I   IF Quantity > 0 THEN PRINT_HW;
2174I End;  { Procedure COMPUTE_CRTS_PTRS }
2175I
2176I
2177I PROCEDURE COMPUTE_CABINETS;
2178I {*****}
2179I { The following routine estimates the number of Patch Panel Cabinets
2180I { and permits the user to increase this for reserve/expansion.
2181I {*****}
2182I
2183I Var
2184I   Config16, Slots, Temp : Integer;
2185I
2186I
2187I Begin { Procedure COMPUTE_CABINETS }
2188I   Temp := Processors;
2189I   {*****}
2190I   { Sufficient system cabinets to house the number of Processors?
2191I   {*****}
2192I   WHILE (Temp MOD 4) > 0 DO
2193I     Temp := Temp + 1;
2194I   SysCab := Temp DIV 4;
2195I   IF (Processors > 0) AND (SysCab = 1) THEN
2196I     SysCab := 1;
2197I   { The following routine estimates the number of PatchPanel Cabinets
2198I   IF SysCab < 1 THEN
2199I     PatchPanel := 1
2200I   ELSE IF SysCab = 0 THEN

```

```

2201I      PatchPanel := 0
2202I      Else IF SysCab > 1 THEN
2203I          PatchPanel := SysCab - 1;
2204I      { The following routine estimates the number of Expansion Cabinets }
2205I      ExpanCab := 0;
2206I      Config16 := Processors DIV 16;
2207I      IF (Processors > (16 * Config16 + 4)) AND
2208I          (Processors < (16 * (Config16 + 1) + 5)) THEN
2209I          ExpanCab := Config16 + 1
2210I      ELSE ExpanCab := Config16;
2211I      IF (Processors > 4) AND (Processors < 21) THEN
2212I          ExpanCab := 1;
2213I      ADDITIONAL_CABINETS;
2214I      Slots := SysCab * 24;
2215I
2216I      { The following permits the user to increase the number of }
2217I      { Patch Panel Cabinets for reserve/expansion. }
2218I      Quantity := PatchPanel + Add_PatchPanel;
2219I      I := I + 1; { I=9 Patch Panel Cabinets }
2220I      IF Quantity > 0 THEN PRINT_HW;
2221I
2222I      { The following permits the user to increase the number of }
2223I      { System Cabinets for reserve/expansion. }
2224I      Quantity := SysCab + Add_System;
2225I      I := I + 1; { I=10 Systems Cabinets }
2226I      IF Quantity > 0 THEN PRINT_HW;
2227I      Quantity := 3 * (SysCab + Add_System); { I=11 Power Modules System Cabinet }
2228I      I := I + 1; { I=11 I/O Power Modules only }
2229I      IF Quantity > 0 THEN PRINT_HW;
2230I
2231I      { The following permits the user to increase the number of }
2232I      { Expansion Cabinets for reserve/expansion. }
2233I      Quantity := ExpanCab + Add_Expansion;
2234I      I := I + 1; { I=12 Expansion Cabinets }
2235I      IF Quantity > 0 THEN PRINT_HW;
2236I      End; { Procedure COMPUTE_CABINETS }
2237I
2238I
2239I Begin { CONFIGURE_PROCESSING_SUBSYSTEM }
2240I     COMPUTE_PROCESSORS;
2241I     COMPUTE_CRTS_PTRS;
2242I     COMPUTE_CABINETS;
2243I End; { CONFIGURE_PROCESSING_SUBSYSTEM }
2244I
2245I
2246I PROCEDURE CONFIGURE_STORAGE_SUBSYSTEM;
2247I
2248I
2249I PROCEDURE COMPUTE_DISK;
2250I

```

```

22511
22521 Var
22531   DiscCtrlr, DiscPatchPnl, THYPERPatchPnl : Integer;
22541
22551 (*****
22561 { The following procedures determine the number of discs, disc
22571 { controllers, disc patch panels, and Patch Panel Cabinets to be ordered }
22581 { The reason that PATCHPNL must be called, which includes THL and ASYNC/
22591 { SYNC routines, from the disc procedure is to maintain the NAVSUP
22601 { required delivery order sequence. Discs are in even quantities due to
22611 { the "mirrored-disc" requirement in SPLICE.
22621 (*****
22631
22641 PROCEDURE COMPUTE_PATCH_PANELS;
22651
22661 Begin { Procedure COMPUTE_PATCH_PANELS }
22671   DiscCtrlr := (D128MB + D240MB + D540MB) DIV 2;
22681   IF (DiscCtrlr MOD 2) > 0 THEN DiscCtrlr := DiscCtrlr + 1;
22691   Quantity := DiscCtrlr;
22701   WHILE (Quantity MOD 4) > 0 DO
22711     Quantity := Quantity + 1;
22721   DiscPatchPnl := Quantity DIV 4; { 4 disc controllers per Disc Patch Panel }
22731   Quantity := DiscPatchPnl;
22741   I := I + 1; { I=13 Disc Patch Panels }
22751   IF Quantity > 0 THEN PRINT_HW;
22761   I := I + 1; { I=14 TANDUM HYPER Link Patch Panels }
22771   IF Stock_Point = 'S' THEN { Is the site a Stock Point site? }
22781     IF (THYPERchannels > 0) and (THYPERchannels < 5) THEN
22791       Begin
22801         Quantity := 1;
22811         PRINT_HW;
22821       End
22831     ELSE
22841       Begin
22851         THYPERPatchPnl := THYPERchannels * 2;
22861         WHILE (THYPERPatchPnl MOD 4) > 0 DO
22871           THYPERPatchPnl := THYPERPatchPnl + 1;
22881         THYPERPatchPnl := THYPERPatchPnl DIV 4;
22891         Quantity := THYPERPatchPnl DIV 4;
22901         IF Quantity > 0 THEN PRINT_HW;
22911       End;
22921   Quantity := AsyncCtrlr;
22931   I := I + 1; { I=15 ASYNC Patch Panels }
22941   IF Quantity > 0 THEN PRINT_HW;
22951   I := I + 1; { I=16 SYNC Patch Panels }
22961   IF Bytesync > 0 THEN
22971     { Only BYTE SYNC lines require SYNC Patch Panels }
22981     Begin
22991       Quantity := Bytesync;
23001       PRINT_HW;

```

```

2301I      End;
2302I End;  ( Procedure COMPUTE_PATCH_PANELS )
2303I
2304I
2305I PROCEDURE COMPUTE_DISK_COMPONENTS;
2306I
2307I
2308I Begin ( Procedure COMPUTE_DISK_COMPONENTS )
2309I   Quantity := DiscCtrlr;
2310I   I := I + 1;                                ( I=17 Disc Controllers )
2311I   IF Quantity > 0 THEN PRINT_HW;
2312I   Quantity := D128MB DIV 2;                    ( Two drawers in each 128MB drive )
2313I   I := I + 1;                                ( I=18 1st Drawer of 128MB Discs )
2314I   IF Quantity > 0 THEN
2315I     Begin
2316I       PRINT_HW;
2317I       I := I + 1;                            ( I=19 2nd Drawer of 128MB Discs )
2318I       PRINT_HW;
2319I     End
2320I   ELSE I := I + 1;
2321I   Quantity := D240MB;
2322I   I := I + 1;                                ( I=20 240MB Discs )
2323I   IF Quantity > 0 THEN PRINT_HW;
2324I   Quantity := D540MB;
2325I   I := I + 1;                                ( I=21 540MB Discs )
2326I   IF Quantity > 0 THEN PRINT_HW;
2327I End;  ( Procedure COMPUTE_DISK_COMPONENTS )
2328I
2329I
2330I Begin ( Procedure COMPUTE_DISK )
2331I   COMPUTE_PATCH_PANELS;
2332I   COMPUTE_DISK_COMPONENTS;
2333I End;  ( Procedure COMPUTE_DISK )
2334I
2335I
2336I PROCEDURE COMPUTE_TAPE;
2337I
2338I { *****
2339I ( This procedure determines the number of Tape Drives and Tape
2340I ( Controllers to be output on the delivery order.
2341I { *****
2342I
2343I Begin ( Procedure COMPUTE_TAPE )
2344I   Quantity := TapeDrv;
2345I   IF Quantity > 0 THEN
2346I     Begin
2347I       I := I + 1;                            ( I=22 Tape Controllers )
2348I       PRINT_HW;
2349I       I := I + 1;                            ( I=23 Tape Drives )
2350I       PRINT_HW;

```

```

2351I      End
2352I      ELSE I := I + 2;
2353I End; { Procedure COMPUTE_TAPE }
2354I
2355I
2356I Begin { Procedure CONFIGURE_STORAGE_SUBSYSTEM }
2357I     COMPUTE_DISK;
2358I     COMPUTE_TAPE;
2359I End; { Procedure CONFIGURE_STORAGE_SUBSYSTEM }
2360I
2361I
2362I PROCEDURE CONFIGURE_INPUT_OUTPUT_SUBSYSTEM;
2363I
2364I
2365I PROCEDURE COMPUTE_READER_PUNCHES;
2366I {*****}
2367I { This procedure determines the number of Reader/Punches and Card Readers }
2368I { to be output on the delivery order. }
2369I {*****}
2370I
2371I Begin { Procedure COMPUTE_READER_PUNCHES }
2372I     Quantity := RdrPunch;
2373I     I := I + 1; { I=24 Card Reader/Punches }
2374I     IF Quantity > 0 THEN PRINT_HW;
2375I     Quantity := CardRdr;
2376I     I := I + 1; { I=25 Card Readers }
2377I     IF Quantity > 0 THEN PRINT_HW;
2378I End; { Procedure COMPUTE_READER_PUNCHES }
2379I
2380I
2381I PROCEDURE COMPUTE_LINE_PRINTERS;
2382I
2383I {*****}
2384I { This procedure determines the number of 1000 LPM and 600 LPM Printers }
2385I { to be output on the delivery order. }
2386I {*****}
2387I
2388I Begin { Procedure COMPUTE_LINE_PRINTERS }
2389I     Quantity := RdrPunch + CardRdr + LPM1000 + LPM600;
2390I     I := I + 1; { I=26 Line Ptr/Crd Rdr Ctrl }
2391I     IF Quantity > 0 THEN PRINT_HW;
2392I     Quantity := LPM1000;
2393I     I := I + 1; { I=27 1000 LPM Printers }
2394I     IF Quantity > 0 THEN PRINT_HW;
2395I     Quantity := LPM600;
2396I     I := I + 1; { I=28 600 LPM Printers }
2397I     IF Quantity > 0 THEN PRINT_HW;
2398I End; { Procedure COMPUTE_LINE_PRINTERS }
2399I
2400I

```

```

2401I Begin ( Procedure CONFIGURE_INPUT_OUTPUT_SUBSYSTEM )
2402I   COMPUTE_READER_PUNCHES;
2403I   COMPUTE_LINE_PRINTERS;
2404I End; ( Procedure CONFIGURE_INPUT_OUTPUT_SUBSYSTEM )
2405I
2406I
2407I PROCEDURE CONFIGURE_COMMUNICATIONS_SUBSYSTEM;
2408I
2409I
2410I PROCEDURE COMPUTE_FOX;
2411I {*****}
2412I { This procedure determines the number of FOX fibre optic controllers and }
2413I { lines to be output on the delivery order. FOX permits SPLICE nodes of }
2414I { 16 OR less Processors (which are co-located within 1000 meters) to be }
2415I { directly interconnected. }
2416I {*****}
2417I
2418I Begin ( Procedure COMPUTE_FOX )
2419I   I := I + 1; { I=29 Skips Interprocessor Bus }
2420I   IF Processors > 16 THEN
2421I     Begin
2422I       I := I + 1; { I=30 FOX CNTRLs for > 16 unit system ! }
2423I       Quantity := Processors; { Processors > 16? If so, order FOX ! }
2424I       WHILE Quantity MOD 16 > 0 DO
2425I         Quantity := Quantity + 1;
2426I       Quantity := Quantity DIV 16;
2427I       PRINT_HW;
2428I       I := I + 1; { I=31 FOX cables }
2429I       Quantity := Quantity - 1;
2430I       PRINT_HW;
2431I     End
2432I   ELSE I := I + 2;
2433I End; ( Procedure COMPUTE_FOX )
2434I
2435I
2436I PROCEDURE COMPUTE_HYPERCHANNELS;
2437I
2438I {*****}
2439I { This procedure is called by COMPUTE_COMMUNICATION_SUBSYSTEM and }
2440I { is invoked only for the configuration of Stock Point Sites. It }
2441I { uses the user inputs for HYPERchannel adapters and connections to }
2442I { write out the correct HYPERchannel component site quantities on }
2443I { the delivery order. Selected componets are written to disk via }
2444I { the PRINT_HW routine. }
2445I {*****}
2446I
2447I
2448I PROCEDURE EXTRA_HYPERCABINETS;
2449I
2450I Begin ( Procedure EXTRA_HYPERCABINETS )

```

```

2451I  Add_HYPERChannel := 0;
2452I  COLOR (15, 1);
2453I  GOTUXY (51, 16);
2454I  WRITE ('HYPERchannel');
2455I
2456I  screen_fieldSS := 1;
2457I  varSS := 1;
2458I  retrieveSS := False;
2459I  last_fieldSS := False;
2460I  retSS := '';
2461I
2462I  IF Stock_Point = 'S' THEN
2463I      GETINT(68,16,2,'N',HYPERCab,'##',0,16,retSS,False,14,1);
2464I
2465I  REPEAT ( until answerSS = 'Y' )
2466I  { Display Items. Change retrieveSS to True and INPUT items}
2467I  REPEAT ( until actionSS = exitSS )
2468I      IF Stock_Point = 'S' THEN
2469I          GETINT(75,16,2,'N',Add_HYPERchannel,'##',0,8,retSS,retrieveSS,14,1);
2470I
2471I          last_fieldSS := True;
2472I          RET_STATUS; { Check the code in "retSS". Set "varSS" and "actionSS" }
2473I
2474I          { Check to see whether to switch retrieveSS to true }
2475I          IF last_fieldSS AND (not retrieveSS) THEN
2476I              Begin
2477I                  retrieveSS := True;
2478I                  last_fieldSS := False;
2479I                  actionSS := staySS;
2480I                  varSS := 1;
2481I              End
2482I          ELSE
2483I              last_fieldSS := False;
2484I          UNTIL actionSS = exitSS;
2485I          ACCEPT_INPUTS;
2486I          UNTIL answerSS = 'Y';
2487I  End; { Procedure EXTRA_HYPERCABINETS }
2488I
2489I
2490I  Begin { Procedure COMPUTE_HYPERCHANNELS }
2491I      A400 := A400 + THYPERchannels;
2492I      { Stores all minicomputer HYPERchannel Adapter requirements }
2493I      Quantity := A400;
2494I      I := I + 1; { I=32 A400 - TANDEM HYPERchannel Adapters }
2495I      IF Quantity > 0 THEN PRINT_HW;
2496I      I := I + 1; { I=33 2nd HYPERchannel Trunk Interface }
2497I      IF Trunks = 2 THEN
2498I          Begin
2499I              Quantity := 1;
2500I              PRINT_HW;

```

```

2501I      End;
2502I      HYPERCab := ((A400 DIV 2) + A150 + AXXX + A220 + A140 + A510) DIV 2;
2503I      EXTRA_HYPERCABINETS;
2504I      Quantity := HYPERCab + Add_HYPERChannel;
2505I
2506I      {*****}
2507I      { The above line determines the number of HYPERchannel cabinets to }
2508I      { be estimated for the user. It assumes that all TANDEM and P-E }
2509I      { HYPERchannels can reside in the same cabinet and that one cabinet }
2510I      { for every two additional adapters will suffice. }
2511I      {*****}
2512I
2513I      I := I + 1;                                { I=34 HYPERchannel Cabinets }
2514I      IF Quantity > 0 THEN PRINT_HW;
2515I      Quantity := THYPERchannels;
2516I      I := I + 1;                                { I=35 THU controllers }
2517I      IF Quantity > 0 THEN PRINT_HW;
2518I      Quantity := Trunks;
2519I      IF Trunks > 0 THEN                          { I=36 - 41 LCN Trunk Line }
2520I      Begin
2521I          Case Cable_Distance of
2522I          'A': Begin
2523I              I := I + 1;                        { I=36 < 500 ft }
2524I              PRINT_HW;
2525I              I := I + 5;
2526I          End;
2527I          'B': Begin
2528I              I := I + 2;                        { I=37 < 1000 ft }
2529I              PRINT_HW;
2530I              I := I + 4;
2531I          End;
2532I          'C': Begin
2533I              I := I + 3;                        { I=38 < 1500 ft }
2534I              PRINT_HW;
2535I              I := I + 3;
2536I          End;
2537I          'D': Begin
2538I              I := I + 4;                        { I=39 < 2500 ft }
2539I              PRINT_HW;
2540I              I := I + 2;
2541I          End;
2542I          'E': Begin
2543I              I := I + 5;                        { I=40 < 4000 ft }
2544I              PRINT_HW;
2545I              I := I + 1;
2546I          End;
2547I          'F': Begin
2548I              I := I + 6;                        { I=41 < 5000 ft }
2549I              PRINT_HW;
2550I          End;

```



```

2551I      End;
2552I      End
2553I      ELSE I := I + 6;
2554I      Quantity := A150;          { A150 - B4800 HYPERchannel Adapter. }
2555I      I := I + 1;              { I=42 HHC1A interfaces }
2556I      IF Quantity > 0 THEN PRINT_HW;
2557I      Quantity := AXXX;          { AXXX - B4900 HYPERchannel Adapter. }
2558I      I := I + 1;              { I=43 DLP interfaces }
2559I      IF Quantity > 0 THEN PRINT_HW;
2560I      Quantity := A150 + AXXX + A220;
2561I      { Burroughs & IBM hosts require ASCII to EBCDIC Conversion Board. }
2562I      I := I + 1;              { I=44 ASCII to EBCDIC Conversion Board }
2563I      IF Quantity > 0 THEN PRINT_HW;
2564I      Quantity := A400 - THYPERchannels; { P-E HYPERchannel Boards }
2565I      I := I + 1;              { I=45 P1 40 Boards for P-E }
2566I      IF Quantity > 0 THEN PRINT_HW;
2567I      Quantity := A220;
2568I      I := I + 1;              { I=46 IBM HYPERchannel Adapters }
2569I      IF Quantity > 0 THEN PRINT_HW;
2570I      Quantity := A140;
2571I      I := I + 1;              { I=47 UNIVAC HYPERchannel Adapters }
2572I      IF Quantity > 0 THEN PRINT_HW;
2573I      Quantity := A510;
2574I      I := I + 1;              { I=48 FIPS HYPERchannel Adapters }
2575I      IF Quantity > 0 THEN PRINT_HW;
2576I      I := I + 1;              { I=49 Find out what line 420301
2577I                               is and insert here }
2578I End; { Procedure COMPUTE_HYPERCHANNELS }
2579I
2580I
2581I PROCEDURE COMPUTE_TERMINAL_COMMUNICATIONS_COMPONENTS;
2582I
2583I Var
2584I   CableOpt, K : Integer;
2585I
2586I
2587I { ***** }
2588I { This procedure is used to handle all SPLICE terminal oriented }
2589I { communications requirements. PRINT_HW is called to write sel- }
2590I { ected components to the output file. }
2591I { ***** }
2592I
2593I Begin { Procedure COMPUTE_TERMINAL_COMMUNICATIONS_COMPONENTS }
2594I   AsyncExtbl := AsyncCtrl * AsyncExtbl;
2595I   IF AsyncCtrl > 0 THEN
2596I     Begin
2597I       Quantity := AsyncCtrl;
2598I       I := I + 1;              { I=40 ASYNC Controllers }
2599I       IF Quantity > 0 THEN PRINT_HW;
2600I       IF AsyncExtbl > 0 THEN

```

APPENDIX B: MAINTENANCE MANUAL Page 86

Page 53

SPLICE.PAS-include file SPLICE2.PAS Program Listing

```

2601I      Begin
2602I      Quantity := AsyncExtbd;
2603I      I := I + 1;      ( I=51 ASYNC Extension Boards )
2604I      PRINT_HW;
2605I      End
2606I      Else I := I + 1;
2607I      End
2608I      Else I := I + 2;
2609I      I := I + 1;      ( I=52 Skips Auto Calling Unit Line Item )
2610I      K := (LIU -1) DIV 45;
2611I      Quantity := LIU;
2612I      IF LIU > 0 THEN
2613I      Begin
2614I      Quantity := K + 1;
2615I      I := I + 1;      ( I=53 6100 Comm Base )
2616I      PRINT_HW;
2617I      IF (LIU > 45*K) AND (LIU <= 45*K+15) THEN
2618I      Begin
2619I      Quantity := 2*K;
2620I      CableOpt := 6*K+2;
2621I      End;
2622I      IF (LIU > 45*K+15) AND (LIU <= 45*K+30) THEN
2623I      Begin
2624I      Quantity := 2*K+1;
2625I      CableOpt := 6*K+4;
2626I      End;
2627I      IF (LIU > 45*K+30) AND (LIU <= 45*(K+1)) THEN
2628I      Begin
2629I      Quantity := 2*(K+1);
2630I      CableOpt := 6*K+6;
2631I      End;
2632I      I := I + 1;      ( I=54 Base ADD-ONS )
2633I      PRINT_HW;
2634I      Quantity := LIU;
2635I      I := I + 1;      ( I=55 LIUs )
2636I      PRINT_HW;
2637I      Quantity := CableOpt;      ( 6100 cables: 2 / base & 2 / add on )
2638I      I := I + 2;      ( Skips 30M & 45M cables )
2639I      I := I + 1;      ( I=58 6100 Cables )
2640I      IF Quantity > 0 THEN PRINT_HW;
2641I      End
2642I      Else I := I + 6;      ( Skips I=53-58 if no 6100 Controllers )
2643I      I := I + 1;      ( I=59 BIT SYNCH Controllers )
2644I      IF BitSynch = 0 THEN
2645I      Begin
2646I      Quantity := BitSynch;
2647I      PRINT_HW;
2648I      End;
2649I      I := I + 1;      ( I=60 BYTE SYNCH Controllers )
2650I      IF ByteSynch > 0 THEN

```

```

2651I      Begin
2652I          Quantity := Bytesync;
2653I          PRINT_HW;
2654I      End;
2655I      I := I + 2;                                ( Skips 1=61-62; ARCLI items ordered)
2656I End;      ( Procedure COMPUTE_TERMINAL_COMMUNICATIONS_COMPONENTS )
2657I
2658I
2659I Begin ( Procedure CONFIGURE_COMMUNICATIONS_SUBSYSTEM )
2660I     COMPUTE_FOX;
2661I     IF Stock_Point = 'S' THEN COMPUTE_HYPERCHANNELS
2662I     Else I := I + 18;
2663I     COMPUTE_TERMINAL_COMMUNICATIONS_COMPONENTS;
2664I End;      ( Procedure CONFIGURE_COMMUNICATIONS_SUBSYSTEM )
2665I
2666I
2667I Begin ( Procedure CONFIGURE_HARDWARE )
2668I     INITIALIZE_HARDWARE_INPUTS;
2669I     GET_HARDWARE_INPUTS;
2670I     CONFIGURE_PROCESSING_SUBSYSTEM;
2671I     CONFIGURE_STORAGE_SUBSYSTEM;
2672I     CONFIGURE_INPUT_OUTPUT_SUBSYSTEM;
2673I     CONFIGURE_COMMUNICATIONS_SUBSYSTEM;
2674I     COMPUTE_SECTION_TOTALS ('Software');
2675I     Mode := Soft;
2676I End;      ( Procedure CONFIGURE_HARDWARE )
2677I
2678I
2679I PROCEDURE CONFIGURE_SOFTWARE;
2680I
2681I
2682I Var
2683I     ( Variables Section For C:SOFTWARE )
2684I     SW6100 : Char;
2685I     ADCCP_6100, ATP_6100, BSC_6100, SNAX_6100, TINET_6100, AM_6520 : String [1];
2686I     DDN, FDC_DLANet, FDC_SNA, NMF_Performance : String [1];
2687I     NMF_Accounting, NMF_Base_Facility, NMF_Diagnostics, NMF_Group : String [1];
2688I     POLL_SELECT, FILE_SECURITY, LCN_FUP, T_TEXT, TR_3271 : String [1];
2689I     NETEX_Months, SPLICE_Net_Months : Integer;
2690I
2691I
2692I (*****)
2693I ( This procedure is used to determine the software requirements for )
2694I ( the delivery order. Please see the rules in the Programmer Main- )
2695I ( tenance Manual to determine which packages are ordered PER )
2696I ( PROCESSOR, PER SITE, and PER PROCESSOR USED. All software )
2697I ( maintenance is PER SITE. Various discount/escalations apply to )
2698I ( the software packages. See the BUILD_COST_TABLE procedure for )
2699I ( specific factors and how they are incorporated into the COSTTABLE )
2700I ( array. )

```

```

2701I {*****}
2702I
2703I
2704I PROCEDURE INITIALIZE_SOFTWARE_INPUTS;
2705I
2706I
2707I Begin { Procedure INITIALIZE SOFTWARE INPUTS }
2708I   { Initialize Variables To Default Values }
2709I   ADCCP_6100 := 'N';
2710I   AM_6520 := 'Y';
2711I   ATP_6100 := 'Y';
2712I   BSC_6100 := 'N';
2713I   DDN := 'N';
2714I   FDC_DLANet := 'N';
2715I   FDC_SNA := 'Y';
2716I   FILE_SECURITY := 'N';
2717I   LCN_FUP := 'N';
2718I   NETEX_Months := 0;
2719I   NMF_Accounting := 'N';
2720I   NMF_Base_Facility := 'N';
2721I   NMF_Diagnostics := 'N';
2722I   NMF_Group := 'N';
2723I   NMF_Performance := 'N';
2724I   POLL_SELECT := 'Y';
2725I   SNAX_6100 := 'Y';
2726I   SPLICENet_Months := 0;
2727I   T_TEXT := 'Y';
2728I   TINET_6100 := 'N';
2729I   TR_3271 := 'N';
2730I End; { Procedure INITIALIZE_SOFTWARE_INPUTS }
2731I
2732I
2733I PROCEDURE GET_SOFTWARE_INPUTS;
2734I
2735I Begin
2736I   screen_fieldSS := 21;
2737I   varSS := 1;
2738I   retrieveSS := False;
2739I   last_fieldSS := False;
2740I   DISPLAY_SCREEN (Screenfile);      { Display Screen }
2741I
2742I   REPEAT { until answerSS = 'Y' }
2743I   { Display Items. Change retrieveSS to True and INPUT items}
2744I   REPEAT { until actionSS = exitSS }
2745I     CASE varSS of
2746I       1: GETITEM(36,5,1,'Y',FILE_SECURITY,'U','','',retSS,retrieveSS,15,1);
2747I       2: IF Stock_Point = 'S' THEN
2748I           GETITEM(36,6,1,'Y',LCN_FUP,'U','','',retSS,retrieveSS,15,1);
2749I       3: GETITEM(36,8,1,'Y',ATP_6100,'U','','',retSS,retrieveSS,15,1);
2750I       4: GETITEM(36,9,1,'Y',BSC_6100,'U','','',retSS,retrieveSS,15,1);

```

```

2751I      5: GETITEM(36,10,1,'Y',ADCCP_6100,'U','','',retSS,retrieveSS,15,1);
2752I      6: GETITEM(36,11,1,'Y',POLL_SELECT,'U','','',retSS,retrieveSS,15,1);
2753I      7: GETITEM(36,12,1,'Y',SNAX_6100,'U','','',retSS,retrieveSS,15,1);
2754I      8: GETITEM(36,13,1,'Y',TINET_6100,'U','','',retSS,retrieveSS,15,1);
2755I      9: GETITEM(36,14,1,'Y',TR_3271,'U','','',retSS,retrieveSS,15,1);
2756I     10: GETITEM(36,15,1,'Y',AM_6520,'U','','',retSS,retrieveSS,15,1);
2757I     11: GETITEM(36,16,1,'Y',T_TEXT,'U','','',retSS,retrieveSS,15,1);
2758I     12: GETITEM(74,5,1,'Y',FDC_SNA,'U','','',retSS,retrieveSS,15,1);
2759I     13: GETITEM(74,6,1,'Y',FDC_DLANet,'U','','',retSS,retrieveSS,15,1);
2760I     14: GETITEM(74,7,1,'Y',DDN,'U','','',retSS,retrieveSS,15,1);
2761I     15: GETITEM(74,9,1,'Y',NMF_Group,'U','','',retSS,retrieveSS,15,1);
2762I     16: IF NMF_Group = 'N' THEN
2763I           GETITEM(74,12,1,'Y',NMF_Base_Facility,
2764I               'U','','',retSS,retrieveSS,15,1);
2765I     17: IF NMF_Group = 'N' THEN
2766I           GETITEM(74,13,1,'Y',NMF_Performance,
2767I               'U','','',retSS,retrieveSS,15,1);
2768I     18: IF NMF_Group = 'N' THEN
2769I           GETITEM(74,14,1,'Y',NMF_Diagnostics,
2770I               'U','','',retSS,retrieveSS,15,1);
2771I     19: IF NMF_Group = 'N' THEN
2772I           GETITEM(74,15,1,'Y',NMF_Accounting,
2773I               'U','','',retSS,retrieveSS,15,1);
2774I     20: IF Stock_Point = 'S' THEN
2775I           GETINT(60,21,2,'N',NETEX_Months,
2776I               '##',0,12,retSS,retrieveSS,15,1);
2777I     21: GETINT(60,22,2,'N',SPLICENet_Months,'##',0,12,retSS,retrieveSS,15,1);
2778I End; ( CASE )
2779I
2780I IF varSS = screen_fieldSS THEN last_fieldSS:=True;
2781I RET_STATUS; ( Check the code in "retSS". Set "varSS" and "actionSS" )
2782I
2783I ( Check to see whether to switch retrieveSS to true )
2784I IF last_fieldSS AND (not retrieveSS) THEN
2785I     Begin
2786I         retrieveSS := True;
2787I         last_fieldSS := False;
2788I         actionSS := staySS;
2789I         varSS := 1;
2790I     End
2791I ELSE
2792I     last_fieldSS := False;
2793I UNTIL actionSS=exitSS;
2794I ACCEPT_INPUTS;
2795I UNTIL answerSS = 'Y';
2796I End; ( Procedure GET_SOFTWARE_INPUTS )
2797I
2798I
2799I PROCEDURE PRINT_SW (Type_Software : Integer);
2800I (*****

```

```

2801I { used in any maintenance computations. }
2802I {*****}
2803I
2804I Begin { Procedure PRINT_SW }
2805I   CASE Type_Software of
2806I     1: Begin { Per Processor Basis }
2807I       Maint_Factor := Momaint_Esc_Rate;
2808I       Extended_Price := Quantity * CostTable[I].purchprice;
2809I     End;
2810I     2: Begin { Per Site Basis }
2811I       Maint_Factor := Momaint_Esc_Rate;
2812I       Extended_Price := CostTable[I].purchprice;
2813I     End;
2814I     3: Begin { Per Processor Basis }
2815I       Maint_Factor := 1;
2816I       Extended_Price := Quantity * CostTable[I].purchprice;
2817I     End;
2818I   End; { End of CASE Statement }
2819I LINE_SETUP;
2820I {*****}
2821I { Compute System Downtime Credit Component Factor per month }
2822I {*****}
2823I   System_Downtime_Component := System_Downtime_Component +
2824I     (Quantity * CostTable[I].basemaint
2825I       * Maint_Factor);
2826I   {*****}
2827I   { Compute the Component Downtime Credit Factor per hour }
2828I   {*****}
2829I   Downtime_Credit := (((CostTable[I].purchprice + CostTable[I].instcost) / 48)
2830I     + (CostTable[I].basemaint * Maint_factor)) * 0.005;
2831I   WRITELN (Diskfile, '', Line_Number:7, ' ', CostTable[I].featureno:8,
2832I     ' ', CostTable[I].descript:28, ' ', Quantity:3,
2833I     CostTable[I].purchprice:13:2, Extended_Price:12:2,
2834I     CostTable[I].basemaint:9:2, Maint_Factor:8:3, Maint_Months:5,
2835I     CostTable[I].basemaint * Maint_Factor * Maint_Months:12:2,
2836I     CostTable[I].instcost:8:2,
2837I     CostTable[I].instcost * Quantity:9:2, Downtime_Credit:9:2,
2838I     (Quantity * CostTable[I].basemaint * Maint_Factor):9:2);
2839I End; { Procedure PRINT_SW }
2840I
2841I
2842I PROCEDURE COMPUTE_PROCESSOR_SOFTWARE;
2843I
2844I Begin { Procedure COMPUTE_PROCESSOR_SOFTWARE }
2845I   Quantity := Processors; { PER PROCESSOR SOFTWARE }
2846I   IF Quantity < 0 THEN
2847I     Begin
2848I       I := 1 + 1; { 1-63 GUARDIAN }
2849I       PRINT_SW (I); { PER PROCESSOR SOFTWARE }
2850I       I := I + 1; { 1-64 BAYNET }

```

```

2851I      PRINT_SW (2);          { PER-SITE SOFTWARE }
2852I      I := I + 1;           { I=65 System Utilities }
2853I      PRINT_SW (2);          { PER-SITE SOFTWARE }
2854I      I := I + 1;           { I=66 ENCOMPASS }
2855I      PRINT_SW (1);          { PER-PROCESSOR SOFTWARE }
2856I      I := I + 5;           { Skips 5 p/o software packages }
2857I      I := I + 1;           { I=72 TPS Software }
2858I      PRINT_SW (2);          { PER-SITE SOFTWARE }
2859I      I := I + 5;           { Skips 5 p/o software packages }
2860I      End
2861I      ELSE I := I + 15;
2862I      I := I + 1;           { I=78 File Security Software }
2863I      IF File_Security = 'Y' THEN PRINT_SW (2);
2864I      I := I + 1;           { I=79 Card Reader Software }
2865I      { PER-SITE SOFTWARE }
2866I      IF CardRdr > 0 THEN PRINT_SW (2);
2867I      I := I + 3;           { Skips 3 p/o software packages }
2868I      End; { Procedure COMPUTE_PROCESSOR_SOFTWARE }
2869I
2870I
2871I      PROCEDURE COMPUTE_COMMUNICATIONS_SOFTWARE;
2872I
2873I      Var
2874I          Temp_Months : Integer;
2875I
2876I
2877I      PROCEDURE COMPUTE_TANDEM_SOFTWARE;
2878I
2879I      Begin { Procedure COMPUTE_TANDEM_SOFTWARE }
2880I          Quantity := Processors;
2881I          IF Quantity > 0 THEN
2882I              Begin
2883I                  I := I + 1;          { I=83 EXPAND Software }
2884I                  PRINT_SW (1);        { PER-PROCESSOR SOFTWARE }
2885I                  I := I + 1;          { I=84 Skips Exchange RJE Software }
2886I          { Possibly need to add choices to software screen for next two items }
2887I                  I := I + 1;          { I=85 AM 3270 Software }
2888I                  PRINT_SW (1);        { PER-PROCESSOR SOFTWARE }
2889I                  I := I + 1;          { I=86 X.25 ACCESS Software }
2890I                  PRINT_SW (1);        { PER-PROCESSOR SOFTWARE }
2891I              End
2892I          ELSE I := I + 4;
2893I          I := I + 1;           { Skips I=87 HYPERLINK Access Method S/W }
2894I          I := I + 1;           { I=88 LCN_FUP }
2895I          IF (SiteInfo.site_type = 'S') AND (LCN_FUP = 'Y') THEN
2896I              PRINT_SW (2);
2897I          I := I + 1;           { I=89 Skip GFE Terminal Support }
2898I          I := I + 1;           { I=90 ATP 6100 }
2899I          IF ATP_6100 = 'Y' THEN PRINT_SW (1);
2900I          I := I + 1;           { I=91 RSC 6100 }

```

APPENDIX B: MAINTENANCE MANUAL Page 92

Page 59

SPLICE.PAS-include file SPLICE2.PAS Program Listing

```

2901I IF BSC_6100 = 'Y' THEN PRINT_SW (1);
2902I I := I + 1; { I=92 ADCCP 6100 }
2903I IF ADCCP_6100 = 'Y' THEN PRINT_SW (1);
2904I I := I + 1; { I=93 TINET 6100 }
2905I IF TINET_6100 = 'Y' THEN PRINT_SW (1);
2906I I := I + 1; { I=94 BURROUGHS POLL-SELECT }
2907I IF POLL_SELECT = 'Y' THEN PRINT_SW (1);
2908I I := I + 1; { I=95 SNAX 6100 }
2909I IF SNAX_6100 = 'Y' THEN PRINT_SW (1);
2910I I := I + 1; { I=96 TR 3271 }
2911I IF TR_3271 = 'Y' THEN PRINT_SW (1);
2912I I := I + 1; { I=97 AM 6520 }
2913I IF AM_6520 = 'Y' THEN PRINT_SW (1);
2914I I := I + 1; { I=98 FDC SNA Interface Package }
2915I IF FDC_SNA = 'Y' THEN PRINT_SW (2); { PER-SITE SOFTWARE }
2916I I := I + 1; { I=99 FDC DLANet Interface Package }
2917I IF FDC_DLANet = 'Y' THEN PRINT_SW (2); { PER-SITE SOFTWARE }
2918I End; { Procedure COMPUTE_TANDEM_SOFTWARE }
2919I
2920I
2921I PROCEDURE COMPUTE_HYPERCHANNEL_SOFTWARE;
2922I
2923I Begin { Procedure COMPUTE_HYPERCHANNEL_SOFTWARE }
2924I I := I + 1; { I=100 HYPERCHANNEL SOFTWARE }
2925I Temp_Months := Maint_Months;
2926I Maint_Months := NETEX_Months;
2927I IF (SiteInfo.site_type = 'S') AND (Temp_Months < Maint_Months)
2928I Begin
2929I Quantity := A150;
2930I PRINT_SW (3);
2931I I := I + 2;
2932I Quantity := 1;
2933I Maint_Months := SPLICE2_Months;
2934I PRINT_SW (2);
2935I End
2936I Else I := I + 2;
2937I Maint_Months := NETEX_Months;
2938I I := I + 1;
2939I IF (SiteInfo.site_type = 'S') AND (Temp_Months < Maint_Months)
2940I Begin
2941I Quantity := AXXX;
2942I PRINT_SW (3);
2943I I := I + 2;
2944I Quantity := 1;
2945I Maint_Months := SPLICE2_Months;
2946I PRINT_SW (2);
2947I End
2948I Else I := I + 2;
2949I Maint_Months := NETEX_Months;
2950I I := I + 1;

```


Page 60

SPLICE.PAS--include file SPLICE2.PAS Program Listing

```

29511 IF (SiteInfo.site_type = 'S') AND ((A400 - THYPERchannels) > 0) THEN
29521   Begin
29531     Quantity := (A400 - THYPERchannels);
29541     PRINT_SW (3);
29551     I := I + 2;           { I=108 CIP, PERKIN-ELMER }
29561     Quantity := 1;
29571     Maint_Months := SPLICENet_Months;
29581     PRINT_SW (2);
29591   End
29601 Else I := I + 2;
29611 Maint_Months := NETEX_Months;
29621 I := I + 1;           { I=109 IBM NETEX Software }
29631 IF (SiteInfo.site_type = 'S') AND (A220 > 0) THEN
29641   Begin
29651     Quantity := A220;
29661     PRINT_SW (3);
29671     I := I + 2;           { I=111 CIP, IBM MVS }
29681     Quantity := 1;
29691     Maint_Months := SPLICENet_Months;
29701     PRINT_SW (2);
29711   End
29721 Else I := I + 2;
29731 Maint_Months := NETEX_Months;
29741 I := I + 1;           { I=112 UNIVAC NETEX Software }
29751 IF (SiteInfo.site_type = 'S') AND (A140 > 0) THEN
29761   Begin
29771     Quantity := A140;
29781     PRINT_SW (3);
29791     I := I + 2;           { I=114 CIP, UNIVAC }
29801     Quantity := 1;
29811     Maint_Months := SPLICENet_Months;
29821     PRINT_SW (2);
29831   End
29841 Else I := I + 2;
29851 Maint_Months := NETEX_Months;
29861 I := I + 1;           { I=115 TANDEM NETEX Software }
29871 IF (SiteInfo.site_type = 'S') AND (THYPERchannels > 0) THEN
29881   Begin
29891     Quantity := THYPERchannels;
29901     PRINT_SW (3);
29911   End;
29921 I := I + 2;           { I=117 CIP, TANDEM }
29931 Quantity := 1;
29941 Maint_Months := SPLICENet_Months;
29951 PRINT_SW (2);
29961 I := I + 1;           { I=118 IBM, TANDEM }
29971 PRINT_SW (2);
29981 End; { Procedure COMPUTE_HYPERCHANNELS_SOFTWARE }
29991
30001

```

```

3001I PROCEDURE COMPUTE_DDN_SOFTWARE;
3002I
3003I Begin { Procedure COMPUTE_DDN_SOFTWARE }
3004I   I := I + 2; { SKIPS TWO OLD DDN PACKAGES }
3005I   Quantity := PROCESSORS; { PER-PROCESSOR SOFTWARE }
3006I   Maint_Months := SPLICE_Net_Months;
3007I   I := I + 1; { I=121 DDN I/F Protocol Software }
3008I   IF DDN = 'Y' THEN PRINT_SW (2); { PER-SITE SOFTWARE }
3009I   Maint_Months := Temp_Months;
3010I   I := I + 1; { I=122 NETWORK MGT FACILITY GROUP }
3011I   IF NMF_Group = 'Y' THEN PRINT_SW (2);
3012I   I := I + 1; { I=123 NMF BASE FACILITY }
3013I   IF NMF_Base_Facility = 'Y' THEN PRINT_SW (2);
3014I   I := I + 1; { I=124 NMF PERFORMANCE MONITORING }
3015I   IF NMF_Performance = 'Y' THEN PRINT_SW (2);
3016I   I := I + 1; { I=125 NMF DIAGNOSTIC MONITORING }
3017I   IF NMF_Diagnostics = 'Y' THEN PRINT_SW (2);
3018I   I := I + 1; { I=126 NMF ACCOUNTING APPLICATION }
3019I   IF NMF_Accounting = 'Y' THEN PRINT_SW (2);
3020I   Quantity := Processors;
3021I   I := I + 2; { Skips 2 p/o software packages }
3022I End; { Procedure COMPUTE_DDN_SOFTWARE }
3023I
3024I Begin { Procedure COMPUTE_COMMUNICATIONS_SOFTWARE }
3025I   COMPUTE_TANDEM_SOFTWARE;
3026I   COMPUTE_HYPERCHANNEL_SOFTWARE;
3027I   COMPUTE_DDN_SOFTWARE;
3028I End; { Procedure COMPUTE_COMMUNICATIONS_SOFTWARE }
3029I
3030I
3031I PROCEDURE COMPUTE_UTILITY_SOFTWARE;
3032I
3033I Begin { Procedure COMPUTE_UTILITY_SOFTWARE }
3034I   I := I + 1; { I=129 File Compression Utility SW }
3035I   IF Processors > 0 THEN PRINT_SW (2); { PER-SITE SOFTWARE }
3036I   Quantity := Processors;
3037I   I := I + 1; { I=130 VAX/VMS Software }
3038I   IF Processors > 0 THEN PRINT_SW (2); { PER-PROCESSOR SOFTWARE }
3039I   I := I + 1; { Skips 2 p/o software packages }
3040I   IF (SiteInfo.sitemo < 2) OR (SiteInfo.sitemo < 3) THEN
3041I     Begin
3042I       I := I + 1; { I=131 PRINT Software }
3043I       IF Quantity > 0 THEN PRINT_SW (2); { PER-PROCESSOR SOFTWARE }
3044I     End
3045I   Else I := I + 1;
3046I   IF (SiteInfo.sitemo < 2) OR (SiteInfo.sitemo < 3) THEN
3047I     Begin
3048I       I := I + 1; { I=132 PERMANENT Software }
3049I       IF Processors > 0 THEN PRINT_SW (2); { PER-PROCESSOR SOFTWARE }
3050I     End

```

Page 62

SPLICE.PAS-include file SPLICE2.PAS Program Listing

```

3051I   Else I := I + 1;
3052I   I := I + 15;                      { Skips 15 p/o software packages. }
3053I   I := I + 1;                      { I=149 TRANSFER }
3054I   IF Processors > 0 THEN PRINT_SW (1); { PER-PROCESSOR Software }
3055I   I := I + 1;                      { I=150 T-TEXT Software }
3056I   IF ('T_TEXT = 'Y') AND (Processors > 0) THEN PRINT_SW (2);
3057I   I := I + 2;                      { Skips two 1 time charge FMSO pkgs }
3058I End;   { Procedure COMPUTE_UTILITY_SOFTWARE }
3059I
3060I Begin { Procedure CONFIGURE_SOFTWARE }
3061I   INITIALIZE_SOFTWARE_INPUTS;
3062I   GET_SOFTWARE_INPUTS;
3063I   COMPUTE_PROCESSOR_SOFTWARE;
3064I   COMPUTE_COMMUNICATIONS_SOFTWARE;
3065I   COMPUTE_UTILITY_SOFTWARE;
3066I   COMPUTE_SECTION_TOTALS ('Documentation');
3067I   Mode := Document;
3068I End;   { PROCEDURE CONFIGURE_SOFTWARE }
3069I
3070I
3071I PROCEDURE INITIALIZE_LAST_SCREEN_DATA;
3072I
3073I Begin { Procedure INITIALIZE_LAST_SCREEN_DATA }
3074I   { Initialize Variables To Default Values }
3075I   Computer_Ops := 0;
3076I   Data_Communication := 0;
3077I   Hardware_Manual := 0;
3078I   Hardware_Overview := 0;
3079I   Operator_Training := 0;
3080I   Per_Call_Months := 3;
3081I   Programmer_Ref := 0;
3082I   Site_Preps := 'N';
3083I   SPLICE_Net_Workshop := 0;
3084I   Sys_Programmer := 0;
3085I   Sys_Resource := 0;
3086I   Sys_Tuning_Xray := 0;
3087I   TAL := 0;
3088I   Training_Group := 5;
3089I End;   { Procedure INITIALIZE_LAST_SCREEN_DATA }
3090I
3091I
3092I PROCEDURE GET_LAST_SCREEN_DATA;
3093I
3094I Begin { Procedure GET_LAST_SCREEN_DATA }
3095I   screen_fieldSS := 14;
3096I   varSS := 1;
3097I   retrieveSS := False;
3098I   last_fieldSS := False;
3099I   DISPLAY_SCREEN (Screenfile);      { Display Screen }
3100I

```

```

31011 REPEAT { until answerSS = 'Y' }
31021 { Display Items. Change retrieveSS to True and INPUT items}
31031 REPEAT { until actionSS = exitSS }
31041 CASE varSS of
31051 1: GETINT(35,5,2,'N',Computer_Ops,'##',0,20,retSS,retrieveSS,15,1);
31061 2: GETINT(35,7,2,'N',Programmer_Ref,'##',0,20,retSS,retrieveSS,15,1);
31071 3: GETINT(35,9,2,'N',Hardware_Manual,'##',0,20,retSS,retrieveSS,15,1);
31081 4: GETINT(35,11,2,'N',Sys_Programmer,'##',0,20,retSS,retrieveSS,15,1);
31091 5: GETINT(70,5,1,'N',Training_Group,'#',1,5,retSS,retrieveSS,15,1);
31101 6: GETINT(75,10,2,'N',Operator_Training,'##',0,20,retSS,retrieveSS,15,1);
31111 7: GETINT(75,11,2,'N',Hardware_Overview,'##',0,20,retSS,retrieveSS,15,1);
31121 8: GETINT(75,12,2,'N',Sys_Resource,'##',0,20,retSS,retrieveSS,15,1);
31131 9: GETINT(75,13,2,'N',Sys_Tuning_Xray,'##',0,20,retSS,retrieveSS,15,1);
31141 10: GETINT(75,14,2,'N',Data_Communication,'##',0,20,retSS,retrieveSS,15,1);
31151 11: GETINT(75,15,2,'N',TAL,'##',0,20,retSS,retrieveSS,15,1);
31161 12: GETINT(75,16,2,'N',SPLICENet_Workshop,'##',0,20,retSS,retrieveSS,15,1);
31171 13: GETINT(35,23,2,'N',Per_Call_Months,'##',0,12,retSS,retrieveSS,15,1);
31181 14: GETITEM(75,23,1,'Y',Site_Preps,'U','','',retSS,retrieveSS,15,1);
31191 End; { CASE }
31201
31211 IF varSS = screen_fieldSS THEN last_fieldSS := True;
31221 RET_STATUS; { Check code in "retSS". Set "varSS" and "actionSS" }
31231
31241 { Check to see whether to switch retrieveSS to true }
31251 IF last_fieldSS AND (not retrieveSS) THEN
31261 Begin
31271 retrieveSS := True;
31281 last_fieldSS := False;
31291 actionSS := staySS;
31301 varSS := 1;
31311 End
31321 ELSE
31331 last_fieldSS := False;
31341 UNTIL actionSS=exitSS;
31351 ACCEPT_INPUTS;
31361 UNTIL answerSS = 'Y';
31371 End; { Procedure GET_LAST_SCREEN_DATA }
31381
31391
31401 PROCEDURE CONFIGURE_DOCUMENTATION;
31411
31421 { *****
31431 { This procedure simply uses the repetitive terminal out procedure }
31441 { MANUAL to list the 4 categories of manuals for the user and asks }
31451 { how many of each should be output on the delivery order. Outputs }
31461 { are written to disk via the PRINT DOC or SEND procedure, }
31471 { described above. Uses WRITE A LINE for actual writes to disk. }
31481 { *****
31491
31501

```

```

31511 Begin ( Procedure CONFIGURE_DOCUMENTATION )
31521   I := I + 1;                                ( I=153 Computer Operations Manual )
31531   Quantity := Computer_Ops;
31541   IF Quantity > 0 THEN PRINT_DOC_OR_TRNG;
31551   I := I + 1;                                ( I=154 Systems Programmer Manual )
31561   Quantity := Sys_Programmer;
31571   IF Quantity > 0 THEN PRINT_DOC_OR_TRNG;
31581   I := I + 1;                                ( I=155 Hardware Manual )
31591   Quantity := Hardware_Manual;
31601   IF Quantity > 0 THEN PRINT_DOC_OR_TRNG;
31611   I := I + 1;                                ( I=156 Programmer Reference Manual )
31621   Quantity := Programmer_Ref;
31631   IF Quantity > 0 THEN PRINT_DOC_OR_TRNG;
31641   COMPUTE_SECTION_TOTALS ('Training');
31651   Mode := Train;
31661 End; ( Procedure CONFIGURE_DOCUMENTATION )
31671
31681
31691 PROCEDURE CONFIGURE_TRAINING;
31701
31711 (*****)
31721 ( This procedure simply uses the repetitive terminal out procedures }
31731 ( GROUP and COURSE to list the 7 categories of courses for the }
31741 ( user and asks which/how many of each should be output on the }
31751 ( delivery order. Outputs are written to disk via the }
31761 ( PRINT_DOC or TRNG procedure, described above. Uses WRITE_A_LINE }
31771 ( for actual writes to disk. }
31781 (*****)
31791
31801 Begin ( Procedure CONFIGURE_TRAINING )
31811   1: Begin
31821     I := I + 1;                                ( I=157 Training Group I )
31831     Quantity := 1;
31841     PRINT_DOC_or_TRNG;
31851     I := I + 1;
31861   End;
31871   2: Begin
31881     I := I + 2;                                ( I=158 Training Group II )
31891     Quantity := 1;
31901     PRINT_DOC_or_TRNG;
31911     I := I + 2;
31921   End;
31931   3: Begin
31941     I := I + 3;                                ( I=159 Training Group III )
31951     Quantity := 1;
31961     PRINT_DOC_or_TRNG;
31971     I := I + 3;
31981   End;
31991   4: Begin

```

APPENDIX B: MAINTENANCE MANUAL Page 98

Page 65

SPLICE.PAS-include file SPLICE2.PAS Program Listing

```

3201I      I := I + 4;      ( I=160 Training Group IV )
3202I      Quantity:= 1;
3203I      PRINT_DOC_or_TRNG;
3204I      End;
3205I      5: I := I + 4;
3206I
3207I      End;
3208I      I := I + 1;      ( I=161 Operator Training Course )
3209I      Quantity := Operator_Training;
3210I      IF Operator_Training > 0 THEN PRINT_DOC_or_TRNG;
3211I      I := I + 1;      ( I=162 Hardware Overview Course )
3212I      Quantity := Hardware_Overview;
3213I      IF Hardware_Overview > 0 THEN PRINT_DOC_or_TRNG;
3214I      I := I + 1;      ( I=163 System Resource Mgmt Course )
3215I      Quantity := Sys_Resource;
3216I      IF Sys_Resource > 0 THEN PRINT_DOC_or_TRNG;
3217I      I := I + 1;      ( I=164 Systems Tuning and XRAY Course )
3218I      Quantity := Sys_Tuning_Xray;
3219I      IF Sys_Tuning_Xray > 0 THEN PRINT_DOC_or_TRNG;
3220I      I := I + 1;      ( I=165 Data Communications Course )
3221I      Quantity := Data_Communication;
3222I      IF Data_Communication > 0 THEN PRINT_DOC_or_TRNG;
3223I      I := I + 1;      ( I=166 TANDEM Application Lang Course )
3224I      Quantity := TAL;
3225I      IF TAL > 0 THEN PRINT_DOC_or_TRNG;
3226I      I := I + 1;      ( I=167 SPLICE.Net WKSHOP )
3227I      Quantity := SPLICE.Net_Workshop;
3228I      IF SPLICE.Net_Workshop > 0 THEN PRINT_DOC_or_TRNG;
3229I      COMPUTE_SECTION_TOTALS ('Maintenance');
3230I      Mode := Maint;
3231I      End; ( Procedure CONFIGURE_TRAINING )
3232I
3233I
3234I      PROCEDURE CONFIGURE_MAINTENANCE;
3235I
3236I      {*****}
3237I      { This procedure is used to write-out the three lines required on }
3238I      { delivery orders for maintenance. Both PM On-Call and On-Call are }
3239I      { written out with Quantity = 1 and all remaining items = 0. The }
3240I      { Maint_Months of Emergency Maintenance are loaded into Quantity }
3241I      { field of the output, multiplied by the updated emergency }
3242I      { maintenance rate and then written to disk. The applicable }
3243I      { uplift rate is written out. All other fields are = 0. }
3244I      {*****}
3245I
3246I
3247I      PROCEDURE PRINT_MAINT;
3248I      {*****}
3249I      { Sets parameters for the three categories to be output on the }
3250I      { delivery order. Sets MONTHS to 0 and Maint_Factor to the }

```

```

3251I { emerg_maint_rate input by the user. Uses WRITE_A_LINE to }
3252I { actually write to disk. }
3253I {*****}
3254I
3255I Begin { FDC Emergency Maint }
3256I   Maint_Months := 0;
3257I   Maint_Factor := Emerg_Maint_Rate;
3258I   Extended_Price := 0;
3259I   WRITE_A_LINE;
3260I End; { Procedure PRINT_MAINT }
3261I
3262I
3263I Begin { Procedure CONFIGURE_MAINTENANCE }
3264I   I := I + 1; { I=168 PM On-Call }
3265I   { If no items have been selected thus far, do not write maintenance
3266I     line items to delivery order output diskfile. }
3267I   IF (Totals [0, 1] > 0) OR (Totals [1, 1] > 0) OR (Totals [2, 1] > 0) OR
3268I     (Totals [3, 1] > 0) OR (Totals [4, 1] > 0) OR (Totals [5, 1] > 0) OR
3269I     (Totals [0, 2] > 0) OR (Totals [1, 2] > 0) OR (Totals [2, 2] > 0) OR
3270I     (Totals [3, 2] > 0) OR (Totals [4, 2] > 0) OR (Totals [5, 2] > 0) THEN
3271I     Begin
3272I       Quantity := 1;
3273I       PRINT_MAINT;
3274I       I := I + 1; { I=169 Skips PM Per-Call Maintenance }
3275I       I := I + 1; { I=170 On-Call Maint }
3276I       PRINT_MAINT;
3277I     End
3278I   ELSE I := I + 2;
3279I   I := I + 1; { I=171 Skips Per-Call Maintenance }
3280I   I := I + 1; { I=172 Emergency Per-Call Maintenance }
3281I   Quantity := Per_Call_Months;
3282I   IF Quantity > 0 THEN PRINT_MAINT;
3283I   COMPUTE_SECTION_TOTALS ('Other');
3284I End; { Procedure CONFIGURE_MAINTENANCE }
3285I
3286I
3287I Begin { Procedure CONFIGURE_COMPONENTS }
3288I   CONFIGURE_HARDWARE;
3289I   CONFIGURE_SOFTWARE;
3290I   INITIALIZE_LAST_SCREEN_DATA;
3291I   GET_LAST_SCREEN_DATA;
3292I   CONFIGURE_DOCUMENTATION;
3293I   CONFIGURE_TRAINING;
3294I   CONFIGURE_MAINTENANCE;
3295I End; { Procedure CONFIGURE_COMPONENTS }
3296I { Name of work procedures include file }
3297I
3298I PROCEDURE SUMMARIZE;
3299I

```

```

3300 Const
3301     LF      : Char = #10;    { Decimal Value for an ASCII line feed  }
3302     CR      : Char = #13;    { Decimal Value for an ASCII carriage return }
3303     Ctrl_Z  : Char = ^Z;     { Value of ASCII "Control-Z" end-of-file marker }
3304
3305 Var
3306     System_Downtime : Real;
3307
3308
3309 Begin { Procedure SUMMARIZE }
3310     System_Downtime := (((Subtotals [0, 1] + Subtotals [0, 3] + Subtotals [1, 1]
3311         + Subtotals [1, 3] + Costtable[1].purchprice)/48)
3312         + System_Downtime_Component) * 0.0125;
3313     WRITELN (Diskfile);
3314     WRITELN (Diskfile);
3315     WRITELN (Diskfile, "NOTES:");
3316     WRITELN (Diskfile);
3317     WRITELN (Diskfile, "", 'MAINTENANCE OPTION = ', SiteInfo.maint_options, '');
3318     WRITELN (Diskfile);
3319     WRITELN (Diskfile, "", 'MAINTENANCE REPAIR AND RESPONSE = ',
3320         SiteInfo.maint_response, '');
3321     WRITELN (Diskfile);
3322     WRITELN (Diskfile,
3323         '"MAINTENANCE REQUIRED FROM END OF NINETY (90) DAY WARRANTY PERIOD."');
3324     WRITELN (Diskfile);
3325     WRITELN (Diskfile,
3326         '"CARD READER AND CARD READER PUNCH CAPABILITIES TEST REQUIREMENTS ARE WAIVED."');
3327     WRITELN (Diskfile);
3328     WRITELN (Diskfile, '"SYSTEM DOWNTIME CREDIT FACTOR PER HOUR EQUALS: $"',
3329         '"w"', '" "', '" "', System_Downtime);
3330     WRITELN (Diskfile);
3331     (*****
3332     (* Terminate the .PRN file with a <CR>, <LF> and *)
3333     (* a <Ctrl Z> End Of File Character. *)
3334     (*****
3335     WRITELN (Diskfile, CR, LF, Ctrl_Z);
3336     CLOSE (Diskfile);
3337     CLOSE (Screenfile);
3338     TextColor (12);
3339     ClrScr;
3340     GOTOXY (4, 9);
3341     WRITELN ('Thank you for using the SPLICE configurer.':58);
3342     WRITELN;
3343     WRITELN;
3344     TextColor (15);
3345     WRITELN ('Your output file is called ':48, PRN_File_Name, '.');
3346     WRITELN;
3347     WRITELN;
3348     TextColor (11);
3349     WRITELN ('The output file is ready for import into LOTUS 1.2.3':65);

```


Page 68

SPLICE.PAS Program Listing

```

3350     TextColor (15);
3351 End;   { Procedure SUMMARIZE }
3352
3353
3354 Begin                                { Main Program }
3355     INITIALIZE;
3356     CONFIGURE_COMPONENTS;
3357     SUMMARIZE;
3358 End.
```

Page 1

CONFMOD.PRG Program Listing

```

1  * PROCEDURE CONFMOD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER THE OPPORTUNITY TO MODIFY OR REVIEW
9  *               ALL DATA IN THE SITE CONFIGURATION DATABASE.
10 *
11 * INPUT FILES   : NONE
12 *
13 * OUTPUT FILE   : NONE
14 *
15 * MODULES CALLED : CONFUPD.PRG. CONFREV.PRG
16 *
17 * CALLED BY     : MAINMENU.PRG
18 *
19 * LOCAL VARIABLES: SELEKT
20 *
21 * DATE LAST TIME MODIFIED =====> 22 DECEMBER 1985 <=====
22 *
23 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE
24 * SELECTION.
25 *
26 STORE "1" TO SELEKT
27 DO WHILE SELEKT < "3"
28     SET COLOR TO W/B, W/B
29     CLEAR
30     ?? FLASH + "W.CONFMOD/"
31     SET CONSOLE OFF
32     WAIT TO SELEKT
33     SET CONSOLE ON
34 *
35 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
36 *
37 DO CASE
38 *
39 *     CALL THE SITE CONFIGURATION UPDATE PROGRAM.
40 CASE SELEKT = "1"
41     DO CONFUPD
42 *
43 *     CALL THE SITE CONDIGURATION REVIEW PROGRAM.
44 CASE SELEKT = "2"
45     DO CONFREV
46 *
47 *     RETURN TO THE MAIN MENU PROGRAM.
48 CASE SELEKT = "3"
49 *
50 ENDCASE

```

Page 2

CONFMOD.PRG Program Listing

```
51 | *
52 | ENDDO WHILE SELEKT < "3"
53 | *
54 | * RETURN TO THE CALLING PROGRAM
55 | *
56 | RETURN
57 | *****
```

```

1  * PROCEDURE CONFREV.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : TO ENABLE THE USER TO REVIEW ANY DATA ELEMENT IN
9  *               THE SITE NAME DATABASE.
10 *
11 * INPUT FILES   : CONFIG.DBF INDICES: CONFIG.NDX
12 *
13 * OUTPUT FILES  : NONE
14 *
15 * CALLED BY     : CONFMOD.PRG
16 *
17 * MODULES CALLED : DELAY.PRG
18 *
19 * GLOBAL VARIABLE: HISITE, LOSITE
20 *
21 * LOCAL VARIABLES: ACCEPT, ANS, CHOICE, ERROR, FIRST_REC, LAST_REC,
22 *                 MADD1, MADD2, MCITY, MCO, MESSAGE, MNAME, MNAMEFL,
23 *                 MOPT, MRESP, MSITE, MSTATE, MTYPE, MZIP
24 *
25 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
26 *
27 * CASE SELECTION = 2      REVIEW EXISTING RECORDS
28 *
29 * USE THE SITE NAME (CONFIG) DATABASE USING THE SITE NUMBER INDEX.
30 *
31 SET ESCAPE OFF
32 SET TALK OFF
33 USE CONFIG
34 GO TOP
35 SET COLOR TO W+/B,W+/B,B
36 CLEAR
37 IF EOF() = .T. THEN
38     SET COLOR TO W+/R, W+/R
39     @ 13,24 SAY " The SITE NAME Database is EMPTY! "
40     DO DELAY
41     RETURN
42 ENDIF
43 ?? FLASH + "S.SITENAME.SCR/"
44 @ 24,0 SAY SPACE (80)
45 SET COLOR TO R+/ ,R+/
46 @ 3,23 SAY " SITE ADDRESS DATA REVIEW FORMAT "
47 STORE "Enter 00 to start at TOF, 99 to start at EOF, or a site " +
48     "number between " + LOSITE + " and " + HISITE + " " TO MESSAGE
49 SET COLOR TO /W, /W
50 @ 24,0 SAY MESSAGE

```

```

51 DO WHILE .T.
52   SET COLOR TO /BR, /BR
53   STORE '00' TO MSITE
54   @ 7,25 GET MSITE PICT '99'
55   READ
56   IF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
57     SET COLOR TO W/B, W/B
58     @ 24,0 SAY SPACE(80)
59     SET COLOR TO W+/R, W+/R
60     STORE ' Response must be between ' + LOSITE + ' and ' +
61       HISITE + ', Zero (00) or 99 ' TO ERROR
62     @ 24,13 SAY ERROR
63     DO DELAY
64     SET COLOR TO /W, /W
65     @ 24,0 SAY MESSAGE
66     LOOP
67   ELSE
68     IF (MSITE = '00' .OR. MSITE = '99') THEN
69       USE CONFIG
70       IF MSITE = '00' THEN
71         GO BOTTOM
72         STORE RECNO() TO LAST_REC
73         GO TOP
74         STORE RECNO() TO FIRST_REC
75       ELSE
76         GO TOP
77         STORE RECNO() TO FIRST_REC
78         GO BOTTOM
79         STORE RECNO() TO LAST_REC
80       ENDIF MSITE = '00'
81       EXIT
82     ELSE
83       USE CONFIG INDEX CONFIG.NDX
84       GO TOP
85       STORE RECNO() TO FIRST_REC
86       GO BOTTOM
87       STORE RECNO() TO LAST_REC
88       FIND &MSITE
89       IF EOF() = .T. THEN
90         SET COLOR TO W/B, W/B
91         @ 24,0 SAY SPACE(80)
92         STORE " No records exist for site number " + MSITE +
93           ", try again " TO ERROR
94         @ 24,16 SAY ERROR
95         SET COLOR TO W+/R, W+/R
96         DO DELAY
97         SET COLOR TO /W, /W
98         @ 24,0 SAY MESSAGE
99         LOOP
100      ELSE

```

Page 3

CONFREV.PRG Program Listing

```

101      EXIT
102      ENDIF EOF() = .T.
103      ENDIF (MSITE = '00' .OR. MSITE = '99')
104  ENDIF
105  ENDDO WHILE .T.
106  *
107  SET COLOR TO W/B, W/B
108  @ 24,0 SAY SPACE(80)
109  *
110  DO WHILE .T.
111      SET COLOR TO R+/B, R+/B
112      @ 5,47 SAY RECNO() PICT "999"
113      SET COLOR TO /BR, /BR
114      @ 7,25 SAY SITENO PICT "99"
115      @ 8,25 SAY SITENAME PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
116      @ 9,25 SAY SITECO PICT "!!!!!!!!!!!!!!!!!!!!!!!!!"
117      @ 10,25 SAY SITENAMEFL PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
118      @ 11,25 SAY SITEADD1 PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
119      @ 12,25 SAY SITEADD2 PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
120      @ 13,25 SAY SITECITY PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
121      @ 14,25 SAY SITESTATE PICT "!!"
122      @ 15,25 SAY SITEZIP PICT "9999999999"
123      @ 16,25 SAY SITETYPE PICT "!!!!"
124      @ 17,35 SAY MAINTOPT PICT "!!!!"
125      @ 18,35 SAY MAINTRESP PICT "!"
126      SET COLOR TO R+/B, R+/B
127      STORE "N" TO CHOICE
128      @ 22,68 GET CHOICE PICT "!"
129      READ
130  *
131  * ENSURE THAT THE USER'S PROMPT IS EITHER "N", "P" OR "X"
132  *
133      DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
134          IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
135              SET COLOR TO W+/R, W+/R
136              @ 24,23 SAY " Response must be either N, P or X "
137              DO DELAY
138              STORE "N" TO CHOICE
139          ENDIF
140          SET COLOR TO R+/B, R+/B
141          @ 22,68 GET CHOICE PICT "!"
142          READ
143      ENDDO
144  *
145  * SKIP TO THE NEXT RECORD TO BE REVIEWED
146  *
147      IF CHOICE = "N" THEN
148          IF RECNO () = LAST_REC THEN
149              GO TOP
150          ELSE

```

```
151         SKIP
152     ENDIF
153 ENDIF
154 *
155 * SKIP TO THE PREVIOUS RECORD TO BE REVIEWED
156 *
157 IF CHOICE = "P" THEN
158     IF RECNO() = FIRST_REC THEN
159         GO BOTTOM
160     ELSE
161         SKIP -1
162     ENDIF
163 ENDIF
164 *
165 * USER HAS DECIDED TO EXIT THE REVIEW
166 *
167 IF CHOICE = "X"
168     EXIT
169 ENDIF
170 *
171 ENDDO WHILE .T.
172 *
173 * RETURN TO CALLING PROGRAM.
174 *
175 RELEASE ALL LIKE M*, ACCEPT, ANS, CHOICE, ERROR, FIRST_REC, LAST_REC
176 CLOSE DATABASES
177 RETURN
178 *****
```

Page 1

CONFUPD.PRG Program Listing

```

1 * PROCEDURE CONFUPD.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. BUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : TO ENABLE THE USER TO MODIFY ANY DATA ELEMENT IN
9 *              : THE SITE NAME DATABASE.
10 *
11 * INPUT FILES  : CONFIG.DBF INDICES: CONFIG.NDX
12 *
13 * OUTPUT FILES : CONFIG.DBF, INDICES: CONFIG.NDX
14 *
15 * MODULES CALLED : DELAY.PRG
16 *
17 * CALLED BY     : CONFMOD.PRG
18 *
19 * GLOBAL VARIABLE: HISITE, LOSITE
20 *
21 * LOCAL VARIABLES: ACCEPT, ANS, CHOICE, ERROR, FIRST_REC, LAST_REC,
22 *                 MADD1, MADD2, MCITY, MCO, MESSAGE, MNAME, MNAMEFL,
23 *                 MOPT, MRESP, MSITE, MSTATE, MTYPE, MZIP, SAVEIT
24 *
25 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
26 *
27 * BEGIN
28 * CASE SELECTION = 1      UPDATE EXISTING RECORDS
29 *
30 * USE THE SITE NAME (CONFIG.DBF) DATABASE USING THE SITE NUMBER INDEX.
31 *
32 SET ESCAPE OFF
33 SET SCOREBOARD OFF
34 SET TALK OFF
35 USE CONFIG
36 GO TOP
37 SET COLOR TO W+/B, W+/B, B
38 CLEAR
39 IF EOF() = .T. THEN
40     SET COLOR TO W+/R, W+/R
41     @ 13,24 SAY " The SITE NAME Database is EMPTY! "
42     DO DELAY
43     RETURN
44 ENDIF
45 ?? FLASH + "S.SITENAME.SCR/"
46 @ 24,0 SAY SPACE(80)
47 SET COLOR TO R+/ ,R+/
48 @ 3,23 SAY " SITE ADDRESS DATA UPDATE FORMAT "
49 STORE "Enter 00 to start at TOP, 99 to start at EOF, or a site " ;
50     "number between " + LOSITE + " and " + HISITE + " " TO MESSAGE

```


Page 2

CONFUPD.PRG Program Listing

```
51 SET COLOR TO /W, /W
52 @ 24,0 SAY MESSAGE
53 DO WHILE .T.
54     SET COLOR TO /BR, /BR
55     STORE '00' TO MSITE
56     @ 7,25 GET MSITE PICT '99'
57     READ
58     IF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
59         SET COLOR TO W/B, W/B
60         @ 24,0 SAY SPACE(80)
61         SET COLOR TO W+/R, W+/R
62         STORE ' Response must be between ' + LOSITE + ' and ' +;
63             HISITE + ', Zero (00) or 99 ' TO ERROR
64         @ 24,13 SAY ERROR
65         DO DELAY
66         SET COLOR TO /W, /W
67         @ 24,0 SAY MESSAGE
68     LOOP
69 ELSE
70     IF (MSITE = '00' .OR. MSITE = '99') THEN
71         USE CONFIG
72         IF MSITE = '00' THEN
73             GO BOTTOM
74             STORE RECNO() TO LAST_REC
75             GO TOP
76             STORE RECNO() TO FIRST_REC
77         ELSE
78             GO TOP
79             STORE RECNO() TO FIRST_REC
80             GO BOTTOM
81             STORE RECNO() TO LAST_REC
82         ENDIF MSITE = '00'
83         EXIT
84     ELSE
85         USE CONFIG INDEX CONFIG.NDX
86         GO TOP
87         STORE RECNO() TO FIRST_REC
88         GO BOTTOM
89         STORE RECNO() TO LAST_REC
90         FIND &MSITE
91         IF EOF() = .T. THEN
92             SET COLOR TO W/B, W/B
93             @ 24,0 SAY SPACE(80)
94             STORE " No records exist for site number " + MSITE +;
95                 ", try again " TO ERROR
96             @ 24,16 SAY ERROR
97             SET COLOR TO W+/R, W+/R
98             DO DELAY
99             SET COLOR TO /W, /W
100            @ 24,0 SAY MESSAGE
```

```

101         LOOP
102         ELSE
103         EXIT
104         ENDIF EOF() = .T.
105         ENDIF (MSITE = '00' .OR. MSITE = '99')
106     ENDIF
107 ENDDO WHILE .T.
108 *
109 SET COLOR TO W/B, W/B
110 @ 24,0 SAY SPACE(80)
111 *
112 STORE SPACE(16) + 'Press "Page Down" key to terminate record update' +;
113     SPACE(16) TO MESSAGE
114 STORE 1 TO INTRO
115 DO WHILE .T.
116     SET COLOR TO /W, /W
117     @ 24,0 SAY MESSAGE
118 *
119 *   INFORM THE USER OF HOW TO TERMINATE THE UPDATE OF A RECORD
120 *
121     IF INTRO = 1 THEN
122         STORE 0 TO INTRO
123         ?? FLASH + "W.CONFUPD/"
124         SET CONSOLE OFF
125         WAIT TO ANS
126         SET CONSOLE ON
127     ENDIF
128 *
129 *   STORING THE OLD RECORD TO A WORK RECORD AREA.  THE M PREFIX
130 *   INDICATES MEMORY VARIABLES DISTINGUISHING THEM FROM THEIR
131 *   CORRESPONDING DATABASE FIELDS.
132 *
133     STORE SITENO      TO MSITE
134     STORE SITENAME    TO MNAME
135     STORE SITECO      TO MCO
136     STORE SITENAMEFL  TO MNAMEFL
137     STORE SITEADD1    TO MADD1
138     STORE SITEADD2    TO MADD2
139     STORE SITECITY    TO MCITY
140     STORE SITESTATE   TO MSTATE
141     STORE SITEZIP     TO MZIP
142     STORE SITETYPE    TO MTYPE
143     STORE MAINTOPT     TO MOPT
144     STORE MAINTRESP    TO MRESP
145 *
146     SET COLOR TO R+/B, R+/B
147     @ 5,47 SAY RECNO() PICT "999"
148     SET COLOR TO /BR, /BR
149 *
150     @ 7,25 SAY MSITE PICT "99"

```

```

151 @ 8,25 GET MNAME PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
152 @ 9,25 GET MCO PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
153 @ 10,25 GET MNAMEFL PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
154 @ 11,25 GET MADD1 PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
155 @ 12,25 GET MADD2 PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
156 @ 13,25 GET MCITY PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
157 @ 14,25 GET MSTATE PICT "!!"
158 @ 15,25 GET MZIP PICT "9999999999"
159 @ 16,25 SAY MTYPE PICT "!!!!"
160 @ 17,35 GET MOPT PICT "!!!!"
161 @ 18,35 GET MRESP PICT "!"
162 READ
163 *
164 * CHECK TO SEE IF ANY RECORD WAS CHANGED
165 *
166 SET COLOR TO W/B, W/B
167 @ 24,0 SAY SPACE(80)
168 STORE 1 TO SAVEIT
169 IF (SITE NO = MSITE)
170     IF (SITE NAME = MNAME)
171         IF (SITE CO = MCO)
172             IF (SITE NAME FL = MNAME FL)
173                 IF (SITE ADD1 = MADD1)
174                     IF (SITE ADD2 = MADD2)
175                         IF (SITE CITY = MCITY)
176                             IF (SITE STATE = MSTATE)
177                                 IF (SITE ZIP = MZIP)
178                                     IF (SITE TYPE = MTYPE)
179                                         IF (MAINT OPT = MOPT)
180                                             IF (MAINT RESP = MRESP)
181                                                 STORE 0 TO SAVEIT
182                                         ENDIF
183                                     ENDIF
184                                 ENDIF
185                             ENDIF
186                         ENDIF
187                     ENDIF
188                 ENDIF
189             ENDIF
190         ENDIF
191     ENDIF
192 ENDIF
193
194 *
195 * ASK THE USER IF HE/SHE DESIRES TO ACCEPT THE CHANGES, ONLY IF ANY
196 * CHANGES WERE MADE
197 *
198 IF SAVEIT = 1 THEN
199     SET COLOR TO W+/B, W+/B
200     @ 20,12 SAY "Do you want to accept the changes? (Yes or No):"

```

Page 5

CONFUPD.PRG Program Listing

```

201 SET COLOR TO R+/B, R+/B
202 @ 20,49 SAY "Y"
203 @ 20,56 SAY "N"
204 STORE "N" TO ACCEPT
205 @ 20,62 GET ACCEPT PICT "!"
206 READ
207 *
208 * ENSURE THAT THE USER'S PROMPT IS EITHER "Y" OR "N"
209 *
210 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
211 IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
212 SET COLOR TO W/B, W/B
213 @ 24,0 SAY SPACE(80)
214 SET COLOR TO W+/R, W+/R
215 @ 24,24 SAY " Response must be either N or Y "
216 DO DELAY
217 STORE "N" TO ACCEPT
218 ENDIF
219 SET COLOR TO R+/B, R+/B
220 @ 20,62 GET ACCEPT PICT "!"
221 READ
222 ENDDO
223 @ 20,10 SAY SPACE (60)
224 *
225 * STORING THE CORRECTED EDIT FIELDS FROM THE WORK AREA.
226 *
227 IF ACCEPT = "Y" THEN
228 REPLACE SITENO WITH MSITE
229 REPLACE SITENAME WITH MNAME
230 REPLACE SITECO WITH MCO
231 REPLACE SITENAMEFL WITH MNAMEFL
232 REPLACE SITEADD1 WITH MADD1
233 REPLACE SITEADD2 WITH MADD2
234 REPLACE SITECITY WITH MCITY
235 REPLACE SITESTATE WITH MSTATE
236 REPLACE SITEZIP WITH MZIP
237 REPLACE SITETYPE WITH MTYPE
238 REPLACE MAINIOPT WITH MOPT
239 REPLACE MAINIRESP WITH MRESP
240 ENDIF
241 ENDIF
242 *
243 SET COLOR TO R+/B, R+/B
244 STORE "N" TO CHOICE
245 @ 22,68 GET CHOICE PICT "!"
246 READ
247 *
248 * FIGURE THAT THE USER'S PROMPT IS EITHER "N", "P" OR "X"
249 *
250 DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")

```

```
251      IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
252          SET COLOR TO W/B, W/B
253          @ 24,0 SAY SPACE(80)
254          SET COLOR TO W+/R, W+/R
255          @ 24,23 SAY " Response must be either N, P or X "
256          DO DELAY
257          STORE "N" TO CHOICE
258      ENDIF
259      SET COLOR TO R+/B, R+/B
260      @ 22,68 GET CHOICE PICT "!"
261      READ
262  ENDDO
263  *
264  * SKIP TO THE NEXT RECORD TO BE REVIEWED .
265  *
266      IF CHOICE = "N" THEN
267          IF RECNO ( ) = LAST_REC THEN
268              GO TOP
269          ELSE
270              SKIP
271          ENDIF
272      ENDIF
273  *
274  * SKIP TO THE PREVIOUS RECORD TO BE REVIEWED
275  *
276      IF CHOICE = "P" THEN
277          IF RECNO ( ) = FIRST_REC THEN
278              GO BOTTOM
279          ELSE
280              SKIP -1
281          ENDIF
282      ENDIF
283  *
284  * USER HAS DECIDED TO EXIT THE REVIEW
285  *
286      IF CHOICE = "X"
287          EXIT
288      ENDIF
289  *
290  ENDDO WHILE .T.
291  *
292  * RETURN TO CALLING PROGRAM.
293  *
294  RELEASE ALL LIKE M*, ACCEPT, ANS, CHOICE, ERROR, FIRST_REC, LAST_REC, SAVEIT
295  CLOSE DATABASES
296  RETURN
297  *****
```

Page 1

DATERPTS.PRG Program Listing

```

1 * PROCEDURE DATERPTS.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              LCDR WINSTON H. BUCKLEY, SC, USN
5 *              LCDR ROBERT F. BRADO, USN
6 *              LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : PROVIDE THE USER A SELECTION OF EFFECTIVE DELIVERY
9 *              ORDER DATE LEVEL REPORTS.
10 *
11 * INPUT FILES  : NONE
12 *
13 * OUTPUT FILES : NONE
14 *
15 * CALLED BY    : REPORCMD.PRG
16 *
17 * MODULES CALLED : EQPDTPRC.PRG, EQPDINPC.PRG, SNODTRPT.PRG
18 *
19 * LOCAL VARIABLES: DATERPTS
20 *
21 * DATE LAST TIME MODIFIED =====> 18 DECEMBER 1985 <=====
22 *
23 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE SELECTION.
24 *
25 STORE "1" TO DATERPTS
26 DO WHILE DATERPTS < "4"
27     SET COLOR TO W/B, W/B, B
28     CLEAR
29     ?? FLASH + "W.DATERPTS/"
30     SET CONSOLE OFF
31     WAIT TO DATERPTS
32     SET CONSOLE ON
33 *
34 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
35 *
36 DO CASE
37 *
38 *     CALL THE EQUIPMENT EFFECTIVE DELIVERY ORDER DATE LEVEL REPORT
39 *     WITH UNIT COST PROGRAM.
40 *     CASE DATERPTS = "1"
41 *         DO EQPDTPRC
42 *
43 *     CALL THE EQUIPMENT EFFECTIVE DELIVERY ORDER DATE LEVEL REPORT
44 *     WITHOUT UNIT COST PROGRAM.
45 *     CASE DATERPTS = "2"
46 *         DO EQPDINPC
47 *
48 *     CALL THE SERIAL NUMBER EFFECTIVE DELIVERY ORDER DATE LEVEL REPORT.
49 *     CASE DATERPTS = "3"
50 *         DO SNODTRPT

```

Page 2

DATERPTS.PRG Program Listing

```
51 | *
52 | *      RETURN TO THE SPLICE REPORTING LEVEL MENU.
53 | *      CASE DATERPTS = "4"
54 | *
55 | *      ENDCASE
56 | *
57 | *      ENDDO (WHILE DATERPTS = "4")
58 | *
59 | *      RETURN TO THE CALLING PROGRAM
60 | *
61 | *      RETURN
62 | *****
```

Page 1

DELAY.PRG Program Listing

```

1 * PROCEDURE DELAY.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              LCDR WINSTON H. BUCKLEY, SC, USN
5 *              LCDR ROBERT J. BRADO, USN
6 *              LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : TO PROVIDE A SHORT DELAY AFTER THE DISPLAY OF AN
9 *              ERROR MESSAGE TO THE USER SUFFICIENT TIME TO READ
10 *             THE MESSAGE.
11 *
12 * INPUT FILES  : NONE
13 *
14 * OUTPUT FILES : NONE
15 *
16 * CALLED BY    : SELECTOR.PRG, MAINMENU.PRG, CONFREV.PRG, CONFUPD.PRG
17 *
18 * LOCAL VARIABLES: DELAY
19 *
20 * DATE LAST TIME MODIFIED =====> 18 DECEMBER 1985 <=====
21 *
22 STORE 1 TO DELAY
23 DO WHILE DELAY < 60
24 *   STORE DELAY + 1 TO DELAY
25 ENDDO DELAY < 60
26 *
27 *   CLEAR OUT THE ERROR MESSAGE
28 *
29 SET COLOR TO W+/B, W+/B
30 @ 24,0 SAY SPACE (80)
31 *
32 *   RETURN TO THE CALLING PROGRAM
33 *
34 RETURN
35 *****

```


Page 1

DESPMOD.PRG Program Listing

```
1  * PROCEDURE DESPMOD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              LCDR WINSTON H. BUCKLEY, SC, USN
5  *              LCDR ROBERT F. BRADO, USN
6  *              LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER THE OPPORTUNITY TO MODIFY OR REVIEW
9  *              ALL DATA IN THE DESCRIPTION DATABASE.
10 *
11 * INPUT FILES   : NONE
12 *
13 * OUTPUT FILE   : NONE
14 *
15 * CALLED BY     : MAINMENU.PRG
16 *
17 * MODULES CALLED : DESPPUPD.PRG, DESPPREV.PRG, DELAY.PRG
18 *
19 * LOCAL VARIABLES: SELEKT
20 *
21 * DATE LAST TIME MODIFIED =====> 22 DECEMBER 1985 <=====
22 *
23 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE
24 * SELECTION.
25 *
26 STORE "1" TO SELEKT
27 DO WHILE SELEKT < "3"
28     SET COLOR TO W/B, W/B
29     CLEAR
30     ?? FLASH + "W.DESPMOD/"
31     SET CONSOLE OFF
32     WAIT TO SELEKT
33     SET CONSOLE ON
34 *
35 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
36 *
37 DO CASE
38 *
39 *     CALL THE DESCRIPTION UPDATE PROGRAM.
40 *     CASE SELEKT = "1"
41 *         DO DESPPUPD
42 *
43 *     CALL THE DESCRIPTION REVIEW PROGRAM.
44 *     CASE SELEKT = "2"
45 *         DO DESPPREV
46 *
47 *     RETURN TO THE MAIN MENU PROGRAM.
48 *     CASE SELEKT = "3"
49 *
50     ENDCASE
```

Page 2

DESPMOD.PRG Program Listing

```
51 | *  
52 | ENDDO (WHILE SELEKT = "3")  
53 | *  
54 | * RETURN TO THE CALLING PROGRAM  
55 | *  
56 | RETURN  
57 | *****
```

Page 1

DESPPREV.PRG Program Listing

```

1  * PROCEDURE DESPPREV.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              : LCDR WINSTON H. BUCKLEY, SC, USN
5  *              : LCDR ROBERT F. BRADO, USN
6  *              : LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : TO ENABLE THE USER TO REVIEW ANY DATA ELEMENT IN
9  *              : THE DESCRIPTION DATABASE.
10 *
11 * INPUT FILES  : DESCRIP.DBF INDICES: DESCRIP.NDX
12 *
13 * OUTPUT FILES : DESCRIP.DBF, INDICES: DESCRIP.NDX
14 *
15 * MODULES CALLED : DELAY.PRG
16 *
17 * CALLED BY    : DESPMOD.PRG
18 *
19 * GLOBAL VARIABLE: LOFNUM, HIFNUM
20 *
21 * LOCAL VARIABLES: ACCEPT, ANS, CHOICE, ERROR, FIRST_REC, LAST_REC,
22 *                MBMAINT, MCLIN, MDESCRIP, MESSAGE, MFLDCMODL,
23 *                MFEAT, MMODELNO, MITCOMP
24 *
25 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
26 *
27 * BEGIN
28 * CASE SELECTION = 2      REVIEW EXISTING RECORDS
29 *
30 * USE DESCRIPTION DATABASE USING THE FEATURE NUMBER INDEX.
31 *
32 SET ESCAPE OFF
33 SET TALK OFF
34 USE DESCRIP
35 GO TOP
36 SET COLOR TO W+/B, W+/B, B
37 CLEAR
38 IF EOF() = .T. THEN
39     SET COLOR TO W+/R, W+/R
40     @ 13,17 SAY " The EQUIPMENT DESCRIPTION Database is EMPTY! "
41     DO DELAY
42     RETURN
43 ENDIF
44 ?? FLASH + "S.DESCRIP.T.SCR/"
45 @ 24,0 SAY SPACE(80)
46 @ 2,39 SAY "REVIEW"
47 STORE ' Enter 00 to start at TOF, 99 to start at EOF, or a six digit ' +;
48 'feature number ' TO MESSAGE
49 SET COLOR TO /W, /W
50 @ 24,0 SAY MESSAGE

```

```

51 DO WHILE .T.
52   SET COLOR TO /BR, /BR
53   STORE '00      ' TO MFEAT
54   @ 6,45 GET MFEAT PICT '999999'
55   READ
56   IF .NOT. ((MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) .OR.;
57             MFEAT = '00      ' .OR. MFEAT = '99      ')
58     SET COLOR TO W/B, W/B
59     @ 24,0 SAY SPACE(80)
60     SET COLOR TO W+/R, W+/R
61     STORE ' Response must be between ' + LOFNUM + ' and ' +;
62           HIFNUM + ', Zero (00) or 99 ' TO ERROR
63     @ 24,8 SAY ERROR
64     DO DELAY
65     SET COLOR TO /W, /W
66     @ 24,0 SAY MESSAGE
67     LOOP
68   ELSE
69     IF (MFEAT = '00      ' .OR. MFEAT = '99      ') THEN
70       USE DESCRIP
71       IF MFEAT = '00      ' THEN
72         GO BOTTOM
73         STORE RECNO() TO LAST_REC
74         GO TOP
75         STORE RECNO() TO FIRST_REC
76       ELSE
77         IF MFEAT = '99      ' THEN
78           GO TOP
79           STORE RECNO() TO FIRST_REC
80           GO BOTTOM
81           STORE RECNO() TO LAST_REC
82         ENDIF MFEAT = '99      '
83       ENDIF MFEAT = '00      '
84       STORE FEATURENO TO MFEAT
85       EXIT
86     ELSE
87       USE DESCRIP INDEX DESCRIP.NDX
88       GO TOP
89       STORE RECNO() TO FIRST_REC
90       GO BOTTOM
91       STORE RECNO() TO LAST_REC
92       FIND &MFEAT
93       IF EOF() = .T. THEN
94         SET COLOR TO W/B, W/B
95         @ 24,0 SAY SPACE(80)
96         SET COLOR TO W+/R, W+/R
97         STORE ' No record exists for feature number ' +;
98               MFEAT + ', try again ' TO ERROR
99         @ 24,12 SAY ERROR
100        DO DELAY

```

```

101         SET COLOR TO /W, /W
102         @ 24,0 SAY MESSAGE
103         LOOP
104     ELSE
105         EXIT
106     ENDIF EOF() = .T.
107     ENDIF (MFEAT = '00' .OR. MFEAT = '99')
108     ENDIF
109     ENDDO WHILE .T.
110     *
111     SET COLOR TO W/B, W/B
112     @ 24,0 SAY SPACE (80)
113     @ 20,20 SAY "To view this field, enter the update mode."
114     DO WHILE .T.
115         SET COLOR TO R+/B, R+/B
116         @ 4,46 SAY RECNO() PICT "99999"
117         SET COLOR TO /BR, /BR
118         @ 6,45 SAY FEATURENO PICT "999999"
119         @ 8,45 SAY CLIN PICT "9999"
120         @ 10,45 SAY DESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
121         @ 12,45 SAY MODELNO PICT "!!!!!!!!!!!!"
122         @ 14,45 SAY FDCMODEL PICT "!!!!!!!!!!!!!!"
123         @ 16,45 SAY TYPECOMPON PICT "!"
124         @ 18,45 SAY BASEMAINT PICT "9999.99"
125         SET COLOR TO R+/B, R+/B
126         STORE "N" TO CHOICE
127         @ 22,67 GET CHOICE PICT "!"
128         READ
129         *
130         * ENSURE THAT THE USER'S PROMPT IS EITHER "N", "P" OR "X"
131         *
132         DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
133             IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
134                 SET COLOR TO W+/R, W+/R
135                 @ 24,23 SAY "Response must be either N, P or X "
136                 DO DELAY
137                 STORE "N" TO CHOICE
138             ENDIF
139             SET COLOR TO R+/B, R+/B
140             @ 22,67 GET CHOICE PICT "!"
141             READ
142         ENDDO
143         *
144         * SKIP TO THE NEXT RECORD TO BE REVIEWED
145         *
146         IF CHOICE = "N" THEN
147             IF RECNO () = LAST_REC THEN
148                 GO TOP
149             ELSE
150                 SKIP

```

Page 4

DESPPREV.PRG Program Listing

```
151         ENDIF
152     ENDIF
153 *
154 * TO THE PREVIOUS RECORD TO BE REVIEWED
155 *
156     IF CHOICE = "p" THEN
157         IF RECNO() = FIRST_REC THEN
158             GO BOTTOM
159         ELSE
160             SKIP -1
161         ENDIF
162     ENDIF
163 *
164 * USER HAS DECIDED TO EXIT THE REVIEW
165 *
166     IF CHOICE = "x"
167         EXIT
168     ENDIF
169 ENDDO WHILE .T.
170 *
171 * RETURN TO CALLING PROGRAM.
172 *
173 RELEASE ALL LIKE M*, CHOICE, ERROR, FIRST_REC, LAST_REC
174 CLOSE DATABASES
175 RETURN
176 *****
```

```

1  * PROCEDURE DESPPUPD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : TO ENABLE THE USER TO MODIFY ANY DATA ELEMENT IN
9  *               THE DESCRIPTION DATABASE.
10 *
11 * INPUT FILES   : DESCRIP.DBF INDICES: DESCRIP.NDX
12 *
13 * OUTPUT FILES  : DESCRIP.DBF, INDICES: DESCRIP.NDX
14 *
15 * MODULES CALLED : DELAY.PRG
16 *
17 * CALLED BY     : DESPMOD.PRG
18 *
19 * GLOBAL VARIABLE: LOFNUM, HIFNUM
20 *
21 * LOCAL VARIABLES: ACCEPT, ANS, CHOICE, ERROR, INTRO, MBMAINT, MCLIN,
22 *                 MDESCRIP, MESSAGE, MFDCMODL, MFEAT, MMODELNO, MITCOMP
23 *
24 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
25 *
26 * BEGIN
27 * CASE SELECTION = 1      UPDATE EXISTING RECORDS
28 *
29 * USE DESCRIPTION DATABASE USING THE FEATURE NUMBER INDEX.
30 *
31 SET ESCAPE OFF
32 SET TALK OFF
33 USE DESCRIP
34 GO TOP
35 SET COLOR TO W+/B, W+/B, B
36 CLEAR
37 IF EOF() = .T. THEN
38     SET COLOR TO W+/R, W+/R
39     @ 13,17 SAY " The EQUIPMENT DESCRIPTION Database is EMPTY! "
40     DO DELAY
41     RETURN
42 ENDIF
43 ?? FLASH + "S.DESCRIP.SCR/"
44 @ 24,0 SAY SPACE(80)
45 @ 2,39 SAY "UPDATE"
46 STORE ' Enter 00 to start at TOP, 99 to start at EOF, or a six digit ' +;
47     'feature number ' TO MESSAGE
48 SET COLOR TO /W, /W
49 @ 24,0 SAY MESSAGE
50 DO WHILE .T.

```

```

51 SET COLOR TO /BR, /BR
52 STORE '00 ' TO MFEAT
53 @ 6,45 GET MFEAT PICT '999999'
54 READ
55 IF .NOT. ((MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) .OR.;
56           MFEAT = '00 ' .OR. MFEAT = '99 ')
57     SET COLOR TO W/B, W/B
58     @ 24,0 SAY SPACE(80)
59     SET COLOR TO W+/R, W+/R
60     STORE ' Response must be between ' + LOFNUM + ' and ' +;
61         HIFNUM + ', Zero (00) or 99 ' TO ERROR
62     @ 24,8 SAY ERROR
63     DO DELAY
64     SET COLOR TO /W, /W
65     @ 24,0 SAY MESSAGE
66     LOOP
67 ELSE
68     IF (MFEAT = '00 ' .OR. MFEAT = '99 ') THEN
69       USE DESCRIP
70       IF MFEAT = '00 ' THEN
71         GO BOTTOM
72         STORE RECNO() TO LAST_REC
73         GO TOP
74         STORE RECNO() TO FIRST_REC
75       ELSE
76         IF MFEAT = '99 ' THEN
77           GO TOP
78           STORE RECNO() TO FIRST_REC
79           GO BOTTOM
80           STORE RECNO() TO LAST_REC
81         ENDIF MFEAT = '99 '
82       ENDIF MFEAT = '00 '
83       STORE FEATURENO TO MFEAT
84       EXIT
85     ELSE
86       USE DESCRIP INDEX DESCRIP.NDX
87       GO TOP
88       STORE RECNO() TO FIRST_REC
89       GO BOTTOM
90       STORE RECNO() TO LAST_REC
91       FIND &MFEAT
92       IF EOF() = .T. THEN
93         SET COLOR TO W/B, W/B
94         @ 24,0 SAY SPACE(80)
95         SET COLOR TO W+/R, W+/R
96         STORE ' No record exists for feature number ' +;
97             MFEAT + ', try again ' TO ERROR
98         @ 24,12 SAY ERROR
99         DO DELAY
100        SET COLOR TO /W, /W

```



```

101          @ 24,0 SAY MESSAGE
102          LOOP
103          ELSE
104          EXIT
105          ENDIF EOF() = .T.
106          ENDIF (MFEAT = '00' ' .OR. MFEAT' = '99' ')
107          ENDIF
108          ENDDO WHILE .T.
109          *
110          STORE SPACE(16) + 'Press "Page Down" key to terminate record update' +;
111              SPACE(16) TO MESSAGE
112          STORE 1 TO INTRO
113          DO WHILE .T.
114          *
115          *   INFORM THE USER OF HOW TO TERMINATE THE UPDATE OF A RECORD
116          *
117          IF INTRO = 1 THEN
118              STORE 0 TO INTRO
119              ?? FLASH + "W.DESPPUPD/"
120              SET CONSOLE OFF
121              WAIT TO ANS
122              SET CONSOLE ON
123          ENDIF
124          *
125          *   STORING THE OLD RECORD TO A WORK RECORD AREA. THE M PREFIX
126          *   INDICATES MEMORY VARIABLES DISTINGUISHING THEM FROM THEIR
127          *   CORRESPONDING DATABASE FIELDS.
128          *
129          STORE FEATURENO TO MFEAT
130          STORE CLIN TO MCLIN
131          STORE DESCRIPT TO MDESCRIP
132          STORE MODELNO TO MMODELNO
133          STORE FDCMODEL TO MFDCMODL
134          STORE TYPECOMPON TO MITCOMP
135          STORE BASEMAINT TO MBMAINT
136          SET COLOR TO R+/B, R+/B
137          @ 4,46 SAY RECNO() PICT "99999"
138          SET COLOR TO /W, /W
139          @ 24,0 SAY MESSAGE
140          *
141          SET COLOR TO /BR, /BR
142          @ 6,45 SAY MFEAT PICT "999999"
143          @ 8,45 GET MCLIN PICT "9999"
144          @ 10,45 GET MDESCRIP PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
145          @ 12,45 GET MMODELNO PICT "!!!!!!!!!!"
146          @ 14,45 GET MFDCMODL PICT "!!!!!!!!!!!!!!"
147          @ 16,45 GET MITCOMP PICT "!"
148          @ 18,45 GET MBMAINT PICT "9999.99"
149          READ
150          *

```

```

151 SET COLOR TO W/B, W/B
152 @ 24,0 SAY SPACE(80)
153 *
154 * ASK THE USER IF HE/SHE DESIRES TO ACCEPT THE CHANGES, ONLY IF ANY
155 * CHANGES WERE MADE
156 *
157 IF .NOT. (FEATURENO=MFEAT .AND. CLIN=MCLIN .AND. DESCRIPT=MDESCRIP .AND.;
158 MODELNO=MMODELNO .AND. FDCMODEL=MFDCMODL .AND. TYPECOMPON =;
159 MICOMP .AND. BASEMAINT=MBMAINT) THEN
160 SET COLOR TO W+/ , W+/
161 @ 21,10 SAY SPACE (55)
162 @ 21,12 SAY "Do you want to accept the changes? (Yes or No):"
163 SET COLOR TO R+/ , R+/
164 @ 21,49 SAY "Y"
165 @ 21,56 SAY "N"
166 STORE "N" TO ACCEPT
167 @ 21,62 GET ACCEPT PICT "!"
168 READ
169 *
170 * ENSURE THAT THE USER'S PROMPT IS EITHER "Y" OR "N"
171 *
172 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
173 IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
174 SET COLOR TO W/B, W/B
175 @ 24,0 SAY SPACE(80)
176 SET COLOR TO W+/R,W+/R
177 @ 24,24 SAY " Response must be either N or Y "
178 DO DELAY
179 STORE "N" TO ACCEPT
180 SET COLOR TO /W, /W
181 @ 24,0 SAY MESSAGE
182 ENDIF
183 SET COLOR TO R+/ ,R+/
184 @ 21,62 GET ACCEPT PICT "!"
185 READ
186 ENDDO
187 SET COLOR TO W+/B, W+/B
188 @ 21,10 SAY SPACE (60)
189 *
190 * STORING THE CORRECTED EDIT FIELDS FROM THE WORK AREA.
191 *
192 IF ACCEPT = "Y" THEN
193 REPLACE FEATURENO WITH MFEAT
194 REPLACE CLIN WITH MCLIN
195 REPLACE DESCRIPT WITH MDESCRIP
196 REPLACE MODELNO WITH MMODELNO
197 REPLACE FDCMODEL WITH MFDCMODL
198 REPLACE TYPECOMPON WITH MICOMP
199 REPLACE BASEMAINT WITH MBMAINT
200 ENDIF

```

```

201   ENDIF
202   *
203   SET COLOR TO W/B, W/B
204   @ 21,10 SAY SPACE (55)
205   *
206   * ASK THE USER IF HE/SHE DESIRES TO CHANGE THE NOTES FIELD
207   *
208   SET COLOR TO W+/B, W+/B
209   @ 20,18 SAY "Edit the NOTES field? (Yes or No):"
210   SET COLOR TO R+/B, R+/B
211   @ 20,42 SAY "Y"
212   @ 20,49 SAY "N"
213   STORE "N" TO ACCEPT
214   @ 20,54 GET ACCEPT PICT "!"
215   READ
216   *
217   * ENSURE THAT THE USER'S PROMPT IS EITHER "Y" OR "N"
218   *
219   DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
220       IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
221           SET COLOR TO W/B, W/B
222           @ 24,0 SAY SPACE(80)
223           SET COLOR TO W+/R, W+/R
224           @ 24,24 SAY " Response must be either N or Y "
225           DO DELAY
226           STORE "N" TO ACCEPT
227           SET COLOR TO /W, /W
228           @ 24,0 SAY MESSAGE
229       ENDIF
230       SET COLOR TO R+/B, R+/B
231       @ 20,54 GET ACCEPT PICT "!"
232       READ
233   ENDDO
234   *
235   IF ACCEPT = "Y" THEN
236       ?? FLASH + "W.NOTES/"
237       SET CONSOLE OFF
238       WAIT TO ANS
239       SET CONSOLE ON
240       CHANGE FIELDS NOTES
241       SET COLOR TO W+/B, W+/B, B
242       CLEAR
243       ?? FLASH + "S.DESCRPT.SCR/"
244       @ 24,0 SAY SPACE(80)
245       SET COLOR TO W+/B, W+/B
246       @ 2,39 SAY "UPDATE"
247       SET COLOR TO R+/B, R+/B
248       @ 4,46 SAY RECNO() PICT "99999"
249       SET COLOR TO /BR, /BR
250       @ 6,45 SAY MFEAT PICT "999999"

```

Page 6

DESPPUPD.PRG Program Listing

```
251      @ 8,45 SAY MCLIN PICT "9999"
252      @ 10,45 SAY MDESCRIP PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
253      @ 12,45 SAY MMODELNO PICT "!!!!!!!!!!!!"
254      @ 14,45 SAY MFDCMODL PICT "!!!!!!!!!!!!!!"
255      @ 16,45 SAY MIYCOMP PICT "!"
256      @ 18,45 SAY MBMAINT PICT "9999.99"
257  ENDIF
258  *
259  SET COLOR TO W/B, W/B
260  @ 20,18 SAY SPACE (50)
261  SET COLOR TO R+/B, R+/B
262  STORE "N" TO CHOICE
263  @ 22,67 GET CHOICE PICT "!"
264  READ
265  *
266  * ENSURE THAT THE USER'S PROMPT IS EITHER "N", "P" OR "X"
267  *
268  DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
269      IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
270          SET COLOR TO W/B, W/B
271          @ 24,0 SAY SPACE(80)
272          SET COLOR TO W+/R, W+/R
273          @ 24,23 SAY " Response must be either N, P or X "
274          DO DELAY
275          STORE "N" TO CHOICE
276      ENDIF
277      SET COLOR TO R+/B, R+/B
278      @ 22,67 GET CHOICE PICT "!"
279      READ
280  ENDDO
281  *
282  * SKIP TO THE NEXT RECORD TO BE REVIEWED
283  *
284  IF CHOICE = "N" THEN
285      IF RECNO ( ) = LAST_REC THEN
286          GO TOP
287      ELSE
288          SKIP
289      ENDIF
290  ENDIF
291  *
292  * SKIP TO THE PREVIOUS RECORD TO BE REVIEWED
293  *
294  IF CHOICE = "P" THEN
295      IF RECNO ( ) = FIRST_REC THEN
296          GO BOTTOM
297      ELSE
298          SKIP -1
299      ENDIF
300  ENDIF
```

```

301 *
302 * HAS DECIDED TO EXIT THE REVIEW
303 *
304     IF CHOICE = "X"
305         EXIT
306     ENDIF
307 ENDDO WHILE .T.
308 *
309 * RETURN TO CALLING PROGRAM.
310 *
311 RELEASE ALL LIKE M*, ACCEPT, ANS, CHOICE, ERROR, INTRO
312 CLOSE DATABASES
313 RETURN
314 *****

```

Page 1

EQPDINPC.PRG Program Listing

```
1 * PROCEDURE EQPDINPC.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. BUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : PROVIDE THE USER A SPLICE EQUIPMENT DELIVERY
9 *              : ORDER DATE LEVEL REPORT WITHOUT UNIT COSTS.
10 *
11 * INPUT FILES  : EQUIP.DBF, EQUIPSD.NDX, DESCRIP.DBF, DESCRIP.NDX,
12 *              : EQUIPSIT.NDX
13 *
14 * OUTPUT FILES : TEMPONE.DBF, TEMPONE.NDX
15 *
16 * MODULES CALLED : DELAY.PRG
17 *
18 * GLOBAL VARIABLE: HIDATE, HISITE, LODATE, LOSITE
19 *
20 * LOCAL VARIABLES: ACCEPT, CHOICE, COLCNT, ERROR, LINECT, MKEY,
21 *              : MNEWDATE, MOLDATE, MSITE, PAGENO, SYSDATE,
22 *              : TODAY, TODATE
23 *
24 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
25 *
26 * CASE SELECTION = 1   EQUIPMENT EFFECTIVE DELIVERY ORDER REPORT
27 *                   WITHOUT UNIT COST
28 *
29 * CALL EQUIPMENT DATABASE INDEXED ON SITE NUMBER.  DISPLAY ALL
30 * EFFECTIVE DATES OF DELIVERY ORDERS FOR THE USER TO SELECT FROM.
31 * CALL EQUIPMENT DATABASE INDEXED ON EFFECTIVE DELIVERY ORDER DATE
32 * AND SITE NUMBER. COPY APPLICABLE RECORDS TO TEMPONE, INDEXED ON
33 * FEATURE NUMBER.  RELATE TO DESCRIPTION FILE.
34 *
35 SET ESCAPE OFF
36 SET TALK OFF
37 SET COLOR TO W+/B, W+/B, B
38 CLEAR
39 USE EQUIP
40 GO TOP
41 IF EOF() = .T. THEN
42     SET COLOR TO W+/R, W+/R
43     @ 13,24 SAY " The EQUIPMENT Database is EMPTY! "
44     DO DELAY
45     RETURN
46 ENDIF
47 ?? FLASH + "S.REPORTS.SCR/"
48 @ 24,0 SAY SPACE(80)
49 SET COLOR TO R+/ , R+/
50 @ 2,25 SAY " DELIVERY ORDER LEVEL REPORT "
```

Page 2

EQPDINPC.PRG Program Listing

```
51 SET COLOR TO W+/BR, W+/BR
52 @ 13,15 SAY "Enter site number for which the report is desired:"
53 *
54 * ENSURE THAT TEMPORARY DATABASE AND INDEX DO NOT EXIST, IF SO ERASE THEM
55 *
56 SET CONSOLE OFF
57 ERASE TEMPONE.DBF
58 ERASE TEMPONE.NDX
59 SET CONSOLE ON
60 USE EQUIP INDEX EQUIPSIT
61 *
62 DO WHILE .T.
63     SET COLOR TO /BR, /BR
64     STORE LOSITE TO MSITE
65     @ 13,66 GET MSITE PICT '99'
66     READ
67     IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
68         SET COLOR TO W+/R, W+/R
69         STORE ' Response must be between ' + LOSITE + ;
70             ' and ' + HISITE + ' ' TO ERROR
71         @ 24,22 SAY ERROR
72         DO DELAY
73         LOOP
74     ELSE
75         GO TOP
76         FIND &MSITE
77         IF EOF() = .T. THEN
78             STORE " No equipment exists for site " + MSITE + ;
79                 ", try another site " TO MESSAGE
80             SET COLOR TO W+/R, W+/R
81             @ 24,15 SAY MESSAGE
82             DO DELAY
83             LOOP
84         ELSE
85             EXIT
86         ENDIF EOF() = .T.
87     ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
88 ENDDO WHILE .T.
89 *
90 SET COLOR TO W+/BR, W+/BR
91 @ 13,15 SAY SPACE(60)
92 *
93 SET COLOR TO W+/B, W+/B
94 @ 05,09 SAY "The following Delivery Order Effective Dates exist for Site"
95 @ 05,69 SAY MSITE
96 SET COLOR TO /BR, /BR
97 @ 13,05 SAY SPACE(70)
98 STORE 1 TO LINECT
99 STORE 1.00 TO COLCNT
100 STORE "000000" TO MOLDATE
```

```

101  *
102  DO WHILE SITENO = MSITE
103      IF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00) THEN
104          @LINECT+6,57 SAY EFFDATE
105      ELSE
106          IF (COLCNT - (COLCNT * (COLCNT/2)) = 0.00) THEN
107              @LINECT+6,38 SAY EFFDATE
108          ELSE
109              @LINECT+6,19 SAY EFFDATE
110          ENDIF (COLCNT - (COLCNT * (COLCNT/2)) = 0.00)
111      ENDIF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00)
112      IF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00) THEN
113          LINECT = 1 + LINECT
114          COLCNT = 1.00
115      ELSE
116          COLCNT = COLCNT + 1.00
117      ENDIF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00)
118      STORE EFFDATE TO MOLDATE
119  *
120  DO WHILE ((EFFDATE = MOLDATE) .AND. .NOT. EOF())
121      SKIP+2
122  ENDDO
123  *
124  IF EOF() THEN
125      EXIT
126  ELSE
127      SKIP
128  ENDIF EOF() = .T.
129  ENDDO WHILE SITENO = MSITE
130  *
131  STORE DIOC(DATE()) TO SYSDATE
132  STORE SUBSTR(SYSDATE,7,2) + SUBSTR(SYSDATE,1,2) + ;
133      SUBSTR(SYSDATE,4,2) TO MDATE
134  STORE SPACE(17) + 'Input Effective Date (Range ' + LODATE + ;
135      ' to ' + HIDATE + ' )' + SPACE(17) TO MESSAGE
136  SET COLOR TO /W, /W
137  @ 24,0 SAY MESSAGE
138  SET COLOR TO W+/B, W+/B
139  @ 3,29 SAY "EFFECTIVE DATE: "
140  *
141  USE EQUIP INDEX EQUIPSD.NDX
142  STORE "000000" TO MOLDATE
143  *
144  DO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDATE)
145      STORE MDATE TO MOLDATE
146      SET COLOR TO R+/B, R+/B
147      @ 3,45 GET MOLDATE PICT "999999"
148      READ
149      DO WHILE .T.
150          IF .NOT. (SUBSTR(MOLDATE,1,2) > "83" .AND. ;

```



```

151         SUBSTR(MOLDATE,1,2) <= "99") THEN
152         SET COLOR TO W/B, W/B
153         @ 24,0 SAY SPACE(80)
154         SET COLOR TO W+/R, W+/R
155         @ 24,16 SAY " Year portion of date must be between 84 and 99 "
156         DO DELAY
157         SET COLOR TO /W, /W
158         @ 24,0 SAY MESSAGE
159         STORE SUBSTR(MDATE,1,2) TO MYEAR
160         SET COLOR TO R+/B, R+/B
161         @ 3,45 GET MYEAR PICT "99"
162         READ
163         STORE MYEAR + SUBSTR(MOLDATE,3,4) TO MOLDATE
164     ELSE
165         EXIT
166     ENDIF
167 ENDDO WHILE .T.
168 *
169 DO WHILE .T.
170     IF .NOT. (SUBSTR(MOLDATE,3,2) >= "01" .AND.;
171             SUBSTR(MOLDATE,3,2) <= "12") THEN
172         SET COLOR TO W/B, W/B
173         @ 24,0 SAY SPACE(80)
174         SET COLOR TO W+/R, W+/R
175         @ 24,16 SAY " Month portion of date must be between 01 and 12 "
176         DO DELAY
177         SET COLOR TO /W, /W
178         @ 24,0 SAY MESSAGE
179         STORE SUBSTR(MDATE,3,2) TO MMONTH
180         SET COLOR TO R+/B, R+/B
181         @ 3,47 GET MMONTH PICT "99"
182         READ
183         STORE SUBSTR(MOLDATE,1,2) + MMONTH +;
184             SUBSTR(MOLDATE,5,2) TO MOLDATE
185     ELSE
186         EXIT
187     ENDIF
188 ENDDO WHILE .T.
189 *
190 DO WHILE .T.
191     IF ((SUBSTR(MOLDATE,3,2) = "04" .OR. SUBSTR(MOLDATE,3,2) = "06" .OR.;
192         SUBSTR(MOLDATE,3,2) = "09" .OR. SUBSTR(MOLDATE,3,2) = "11") .AND.;
193         .NOT. (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
194             SUBSTR(MOLDATE,5,2) <= "30")) THEN
195         SET COLOR TO W/B, W/B
196         @ 24,0 SAY SPACE(80)
197         SET COLOR TO W+/R, W+/R
198         @ 24,16 SAY " Day portion of date must be between 01 and 30 "
199         DO DELAY
200         SET COLOR TO /W, /W

```

```

201      @ 24,0 SAY MESSAGE
202      STORE SUBSTR(MDATE,5,2) TO MDAY
203      SET COLOR TO R+/B, R+B
204      @ 3,49 GET MDAY PICT "99"
205      READ
206      STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
207      LOOP
208  ELSE
209  *
210      IF (SUBSTR(MOLDATE,3,2) = "02" .AND. .NOT.;
211      (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
212      SUBSTR(MOLDATE,5,2) <= "28")) THEN
213      SET COLOR TO W/B, W/B
214      @ 24,0 SAY SPACE(80)
215      SET COLOR TO W+/R, W+/R
216      @ 24,16 SAY " Day portion of date must be between 01 and 28 "
217      DO DELAY
218      SET COLOR TO /W, /W
219      @ 24,0 SAY MESSAGE
220      STORE SUBSTR(MDATE,5,2) TO MDAY
221      SET COLOR TO R+/B, R+B
222      @ 3,49 GET MDAY PICT "99"
223      READ
224      STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
225      LOOP
226  ELSE
227  *
228      IF .NOT. (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
229      SUBSTR(MOLDATE,5,2) <= "31") THEN
230      SET COLOR TO W/B, W/B
231      @ 24,0 SAY SPACE(80)
232      SET COLOR TO W+/R, W+/R
233      @ 24,16 SAY " Day portion of date must be between 01 and 31 "
234      DO DELAY
235      SET COLOR TO /W, /W
236      @ 24,0 SAY MESSAGE
237      STORE SUBSTR(MDATE,5,2) TO MDAY
238      SET COLOR TO R+/B, R+B
239      @ 3,49 GET MDAY PICT "99"
240      READ
241      STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
242      LOOP
243  ELSE
244      EXIT
245  ENDIF
246  ENDIF
247  ENDIF
248  ENDLO WHILE .T.
249  *
250  GO TOP

```

```

251 FIND &MOLDATE
252 IF EOF() = .T. THEN
253     SET COLOR TO W/B, W/B
254     @ 24,0 SAY SPACE(80)
255     STORE "EFFECTIVE DATE " + MOLDATE + " does not exist for site " +
256         MSITE + ", try another " TO NODATE
257     SET COLOR TO W+/R, W+/R
258     @ 24,06 SAY NODATE
259     DO DELAY
260     SET COLOR TO /W, /W
261     @ 24,0 SAY MESSAGE
262     STORE "000000" TO MOLDATE
263     LOOP
264     ENDIF EOF() = .T.
265 ENDDO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDATE)
266 *
267 SET COLOR TO W+/B, W+/B
268 @ 05,05 SAY SPACE(70)
269 @ 24,0 SAY SPACE(80)
270 *
271 * CLEAR LISTING OF EFFECTIVE DATES FROM SCREEN
272 *
273 SET COLOR TO /BR, /BR
274 @ 07,2 SAY SPACE(76)
275 @ 08,2 SAY SPACE(76)
276 @ 09,2 SAY SPACE(76)
277 @ 10,2 SAY SPACE(76)
278 @ 11,2 SAY SPACE(76)
279 @ 12,2 SAY SPACE(76)
280 @ 13,2 SAY SPACE(76)
281 @ 14,2 SAY SPACE(76)
282 @ 15,2 SAY SPACE(76)
283 @ 16,2 SAY SPACE(76)
284 @ 17,2 SAY SPACE(76)
285 @ 18,2 SAY SPACE(76)
286 @ 19,2 SAY SPACE(76)
287 @ 20,2 SAY SPACE(76)
288 @ 21,2 SAY SPACE(76)
289 *
290 SET COLOR TO R+/ , R+/
291 @ 13,18 SAY " CREATING TEMPORARY DATABASE AND INDEX FILE "
292 STORE "MOLDATE" + "MSITE" TO MKEY
293 GO TOP
294 FIND &MKEY
295 *
296 COPY TO TEMPONE FOR SITE NO = "&MSITE" .AND. EFFDATE = "&MOLDATE"
297 SELECT 1
298 USE TEMPONE
299 INDEX ON FEATURENO TO TEMPONE
300 SELECT 2

```

```

301 USE DESCRIP INDEX DESCRIP
302 SELECT TEMPONE
303 SET RELATION TO FEATURENO INTO DESCRIP
304 GO TOP
305 *
306 *   CREATE THE SPLICE EQUIPMENT PROJECT REPORT AND CHECK IF THE REPORT
307 *   IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
308 *
309 SET COLOR TO W+/BR, W+/BR
310 @ 13,15 SAY SPACE(60)
311 @ 13,16 SAY " Do you want a printed report? (Yes or No): "
312 SET COLOR TO /BR, /BR
313 @ 13,49 SAY "Y"
314 @ 13,56 SAY "N"
315 STORE "N" TO ACCEPT
316 @ 13,62 GET ACCEPT PICT "!"
317 READ
318 *
319 *   ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
320 *
321 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
322   IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
323     SET COLOR TO W+/R, W+/R
324     @ 24,24 SAY " Response must be either N or Y "
325     DO DELAY
326     STORE "N" TO ACCEPT
327   ENDIF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
328   SET COLOR TO /BR, /BR
329   @ 13,62 GET ACCEPT PICT "!"
330   READ
331 ENDDO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
332 *
333 SET COLOR TO /BR, /BR
334 @ 13,15 SAY SPACE(55)
335 *
336 IF ACCEPT = "Y" THEN
337   ?? FLASH + "W.PRINTER/"
338   SET CONSOLE OFF
339   WAIT TO CHOICE
340   SET CONSOLE ON
341   SET COLOR TO W/B, W/B
342   @ 22,10 SAY SPACE(65)
343   STORE DIOC( DATE() ) TO TODAY
344   STORE SUBSTR( TODAY, 4, 2 ) + " " + CMONTH( DATE() ) + " 19" + ;
345     SUBSTR( TODAY, 7, 2 ) TO TODATE
346   STORE 0 TO PAGENO
347   STORE 61 TO LINECT
348   SET COLOR TO R+/ , R+/
349   SET DEVICE TO PRINT
350 *

```

```

351 DO WHILE .NOT. EOF()
352     DO WHILE (LINECT <= 60 .AND. .NOT. EOF())
353         @ LINECT,3 SAY SITENO PICT "99"
354         @ LINECT,9 SAY B->CLIN PICT "9999"
355         @ LINECT,17 SAY FEATURENO PICT "999999"
356         @ LINECT,28 SAY B->DESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
357         @ LINECT,60 SAY QTY PICT "999"
358         @ LINECT,67 SAY B->FDCMODEL PICT "!!!!!!!!!!!!!!"
359         LINECT = LINECT + 1
360         SKIP
361     ENDDO WHILE (LINECT <= 60 .AND. .NOT. EOF())
362 *
363     IF EOF() = .T. THEN
364         IF PAGENO > 1 THEN
365             @ 62,37 SAY "Page " + STR(PAGENO,2,0)
366         ENDIF PAGENO > 1
367         EJECT
368         SET DEVICE TO SCREEN
369         @ 13,25 SAY " FINISHED PRINTING THE REPORT "
370         DO DELAY
371         EXIT
372     ELSE
373         SET DEVICE TO SCREEN
374         @ 13,27 SAY " Printing Page Number " + STR(PAGENO + 1,2,0) + " "
375         SET DEVICE TO PRINT
376     ENDIF EOF() = .T.
377 *
378     IF (LINECT > 60 .AND. PAGENO > 1) THEN
379         @ 62,37 SAY "Page " + STR(PAGENO,2,0)
380     ENDIF (LINECT > 60 .AND. PAGENO > 1)
381     @ 2,25 SAY " DELIVERY ORDER LEVEL REPORT "
382     @ 3,29 SAY "EFFECTIVE DATE: "
383     @ 3,45 SAY MOLDATE
384     @ 4,60 SAY TODATE
385     @ 6,2 SAY "SITE CLIN FEATURE# DESCRIPTION"
386     @ 6,60 SAY "QTY MODEL NUMBER"
387     @ 7,2 SAY "===== "
388     @ 7,51 SAY "===== "
389     PAGENO = PAGENO + 1
390     STORE 9 TO LINECT
391 *
392     ENDDO WHILE .NOT. EOF()
393 ELSE
394     SET COLOR TO GR+/B, GR+/B
395     @ 5,2 SAY "SITE CLIN FEATURE# DESCRIPTION"
396     @ 5,60 SAY "QTY MODEL NUMBER"
397     SET COLOR TO /BR, /BR
398     STORE 0 TO LINECT
399 *
400 DO WHILE .NOT. EOF()

```

```

401 DO WHILE LINECT < 15
402   @ LINECT+7,3 SAY SITENO PICT "99"
403   @ LINECT+7,9 SAY B->CLIN PICT "9999"
404   @ LINECT+7,17 SAY FEATURENO PICT "999999"
405   @ LINECT+7,28 SAY B->DESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!"
406   @ LINECT+7,60 SAY QTY PICT "999"
407   @ LINECT+7,67 SAY B->FDCMODEL PICT "!!!!!!!!!!!"
408   LINECT = LINECT + 1
409   SKIP
410   IF EOF() = .T. THEN
411     SET COLOR TO W+/R, W+/R
412     @ 24,18 SAY " End of File reached, Press any key to EXIT "
413     SET CONSOLE OFF
414     WAIT TO ACCEPT
415     SET CONSOLE ON
416     EXIT
417   ENDIF EOF() = .T.
418 ENDDO WHILE LINECT < 15
419 *
420 IF EOF() = .T. THEN
421   EXIT
422 ENDIF EOF() = .T.
423 SET COLOR TO R+/B, R+/B
424 STORE "C" TO CHOICE
425 @ 22,57 GET CHOICE PICT "!"
426 READ
427 *
428 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
429 *
430 DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
431   IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
432     SET COLOR TO W+/R, W+/R
433     @ 24,24 SAY " Response must be either C or X "
434     DO DELAY
435     STORE "C" TO CHOICE
436   ENDIF .NOT. (CHOICE = "C" .OR. CHOICE = "X")
437   SET COLOR TO R+/B, R+/B
438   @ 22,57 GET CHOICE PICT "!"
439   READ
440 ENDDO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
441 *
442 * DETERMINE IF THE USER WANTS TO QUIT OR CONTINUE
443 *
444 IF CHOICE = "C"
445   SET COLOR TO /BR, /BR
446   @ 07,2 SAY SPACE(76)
447   @ 08,2 SAY SPACE(76)
448   @ 09,2 SAY SPACE(76)
449   @ 10,2 SAY SPACE(76)
450   @ 11,2 SAY SPACE(76)

```

```
451      @ 12,2 SAY SPACE(76)
452      @ 13,2 SAY SPACE(76)
453      @ 14,2 SAY SPACE(76)
454      @ 15,2 SAY SPACE(76)
455      @ 16,2 SAY SPACE(76)
456      @ 17,2 SAY SPACE(76)
457      @ 18,2 SAY SPACE(76)
458      @ 19,2 SAY SPACE(76)
459      @ 20,2 SAY SPACE(76)
460      @ 21,2 SAY SPACE(76)
461      STORE 0 TO LINECT
462      ELSE
463          EXIT
464      ENDIF CHOICE = "C"
465      *
466      ENDDO WHILE .NOT. EOF()
467      *
468      ENDIF ACCEPT = "Y"
469      *
470      *   ERASE ALL TEMPORARY FILES CREATED DURING REPORT GENERATION
471      *
472      CLOSE DATABASES
473      SET CONSOLE OFF
474      ERASE TEMPONE.DBF
475      ERASE TEMPONE.NDX
476      SET CONSOLE ON
477      SET PRINT OFF
478      *
479      *   RETURN TO CALLING PROGRAM
480      *
481      RELEASE ALL LIKE M*, ACCEPT, CHOICE, COLCNT, ERROR, LINECT, PAGENO,;
482          SYSDATE, TODAY, TODAY
483      RETURN
484      *****
```

Page 1

EQPDTPRC.PRG Program Listing

```
1 * PROCEDURE EQPDTPRC.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. BUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : PROVIDE THE USER A SPLICE EQUIPMENT DELIVERY
9 *              : ORDER DATE LEVEL REPORT WITH UNIT COSTS.
10 *
11 * INPUT FILES  : EQUIP.DBF, EQUIPSD.NDX, DESCRIP.DBF, DESCRIP.NDX,
12 *              : EQUIPSIT.NDX
13 *
14 * OUTPUT FILES : TEMPONE.DBF, TEMPONE.NDX
15 *
16 * MODULES CALLED : DELAY.PRG
17 *
18 * GLOBAL VARIABLE: HIDATE, HISITE, LODATE, LOSITE
19 *
20 * LOCAL VARIABLES: ACCEPT, CHOICE, COLCNT, ERROR, LINECT, MKEY, MNEWDATE,
21 *              : MOLDATE, MSITE, PAGENO, SYSDATE, TODAY, TODATE
22 *
23 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
24 *
25 * CASE SELECTION = 1    EQUIPMENT EFFECTIVE DELIVERY ORDER REPORT
26 *                      : WITH UNIT COST
27 *
28 * CALL EQUIPMENT DATABASE INDEXED ON SITE NUMBER.  DISPLAY ALL
29 * EFFECTIVE DATES OF DELIVERY ORDERS FOR THE USER TO SELECT FROM.
30 * CALL EQUIPMENT DATABASE INDEXED ON EFFECTIVE DELIVERY ORDER DATE
31 * AND SITE NUMBER. COPY APPLICABLE RECORDS TO TEMPONE, INDEXED ON
32 * FEATURE NUMBER. RELATE TO DESCRIPTION FILE.
33 *
34 SET ESCAPE OFF
35 SET TALK OFF
36 SET COLOR TO W+/B, W+/B, B
37 CLEAR
38 USE EQUIP
39 GO TOP
40 IF EOF() = .T. THEN
41     SET COLOR TO W+/R, W+/R
42     @ 13,24 SAY " The EQUIPMENT Database is EMPTY! "
43     DO DELAY
44     RETURN
45 ENDIF
46 ?? FLASH + "S.REPORTS.SCR/"
47 @ 24,0 SAY SPACE(80)
48 SET COLOR TO R+/ , R+/
49 @ 2,25 SAY " DELIVERY ORDER LEVEL REPORT "
50 SET COLOR TO W+/BR, W+/BR
```



```

51 @ 13,15 SAY "Enter site number for which the report is desired:"
52 *
53 * ENSURE THAT TEMPORARY DATABASE AND INDEX DO NOT EXIST, IF SO ERASE THEM
54 *
55 SET CONSOLE OFF
56 ERASE TEMPONE.DBF
57 ERASE TEMPONE.NDX
58 SET CONSOLE ON
59 USE EQUIP INDEX EQUIPSIT
60 *
61 DO WHILE .T.
62     SET COLOR TO /BR, /BR
63     STORE LOSITE TO MSITE
64     @ 13,66 GET MSITE PICT '99'
65     READ
66     IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
67         SET COLOR TO W+/R, W+/R
68         STORE ' Response must be between ' + LOSITE + ;
69             ' and ' + HISITE + ' ' TO ERROR
70         @ 24,22 SAY ERROR
71         DO DELAY
72         LOOP
73     ELSE
74         GO TOP
75         FIND &MSITE
76         IF EOF() = .T. THEN
77             STORE " No equipment exists for site " + MSITE + ;
78                 ", try another site " TO MESSAGE
79             SET COLOR TO W+/R, W+/R
80             @ 24,15 SAY MESSAGE
81             DO DELAY
82             LOOP
83         ELSE
84             EXIT
85         ENDIF EOF() = .T.
86     ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
87 ENDDO WHILE .T.
88 *
89 SET COLOR TO W+/BR, W+/BR
90 @ 13,15 SAY SPACE(60)
91 *
92 SET COLOR TO W+/B, W+/B
93 @ 05,09 SAY "The following Delivery Order Effective Dates exist for Site"
94 @ 05,69 SAY MSITE
95 SET COLOR TO /BR, /BR
96 @ 13,05 SAY SPACE(70)
97 STORE 1 TO LINECT
98 STORE 1.00 TO COLCNT
99 STORE "000000" TO MOLDATE
100 *

```

Page 3

EQPDTPRC.PRG Program Listing

```
101 DO WHILE SITENO = MSITE
102   IF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00) THEN
103     @LINECT+6,57 SAY EFFDATE
104   ELSE
105     IF (COLCNT - (COLCNT * (COLCNT/2)) = 0.00) THEN
106       @LINECT+6,38 SAY EFFDATE
107     ELSE
108       @LINECT+6,19 SAY EFFDATE
109     ENDIF (COLCNT - (COLCNT * (COLCNT/2)) = 0.00)
110   ENDIF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00)
111   IF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00) THEN
112     LINECT = 1 + LINECT
113     COLCNT = 1.00
114   ELSE
115     COLCNT = COLCNT + 1.00
116   ENDIF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00)
117   STORE EFFDATE TO MOLDATE
118 *
119   DO WHILE ((EFFDATE = MOLDATE) .AND. .NOT. EOF())
120     SKIP+2
121   ENDDO
122 *
123   IF EOF() THEN
124     EXIT
125   ELSE
126     SKIP
127   ENDIF EOF() = .T.
128 ENDDO WHILE SITENO = MSITE
129 *
130 STORE DTOC(DATE()) TO SYSDATE
131 STORE SUBSTR(SYSDATE,7,2) + SUBSTR(SYSDATE,1,2) +;
132   SUBSTR(SYSDATE,4,2) TO MDATE
133 STORE SPACE(17) + 'Input Effective Date (Range ' + LODATE +;
134   ' to ' + HIDATE + ')' + SPACE(17) TO MESSAGE
135 SET COLOR TO /W, /W
136 @ 24,0 SAY MESSAGE
137 SET COLOR TO W+/B, W+/B
138 @ 3,29 SAY "EFFECTIVE DATE: "
139 *
140 USE EQUIP INDEX EQUIPSD.NDX
141 STORE "000000" TO MOLDATE
142 *
143 DO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDATE)
144   STORE MDATE TO MOLDATE
145   SET COLOR TO R+/B, R+/B
146   @ 3,45 GET MOLDATE PICT "999999"
147   READ
148   DO WHILE .T.
149     IF .NOT. (SUBSTR(MOLDATE,1,2) > "83" .AND.;
150       SUBSTR(MOLDATE,1,2) <= "99") THEN
```

```
151      SET COLOR TO W/B, W/B
152      @ 24,0 SAY SPACE(80)
153      SET COLOR TO W+/R, W+/R
154      @ 24,16 SAY " Year portion of date must be between 84 and 99 "
155      DO DELAY
156      SET COLOR TO /W, /W
157      @ 24,0 SAY MESSAGE
158      STORE SUBSTR(MDATE,1,2) TO MYEAR
159      SET COLOR TO R+/B, R+/B
160      @ 3,45 GET MYEAR PICT "99"
161      READ
162      STORE MYEAR + SUBSTR(MOLDATE,3,4) TO MOLDATE
163      ELSE
164      EXIT
165      ENDIF
166      ENDDO
167      *
168      DO WHILE .T.
169      IF .NOT. (SUBSTR(MOLDATE,3,2) > "00" .AND.;
170      SUBSTR(MOLDATE,3,2) < "13") THEN
171      SET COLOR TO W/B, W/B
172      @ 24,0 SAY SPACE(80)
173      SET COLOR TO W+/R, W+/R
174      @ 24,16 SAY " Month portion of date must be between 01 and 12 "
175      DO DELAY
176      SET COLOR TO /W, /W
177      @ 24,0 SAY MESSAGE
178      STORE SUBSTR(MDATE,3,2) TO MMONTH
179      SET COLOR TO R+/B, R+/B
180      @ 3,47 GET MMONTH PICT "99"
181      READ
182      STORE SUBSTR(MOLDATE,1,2) + MMONTH +;
183      SUBSTR(MOLDATE,5,2) TO MOLDATE
184      ELSE
185      EXIT
186      ENDIF
187      ENDDO
188      *
189      DO WHILE .T.
190      IF ((SUBSTR(MOLDATE,3,2) = "04" .OR. SUBSTR(MOLDATE,3,2) = "06" .OR.;
191      SUBSTR(MOLDATE,3,2) = "09" .OR. SUBSTR(MOLDATE,3,2) = "11") .AND.;
192      .NOT. (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
193      SUBSTR(MOLDATE,5,2) <= "30")) THEN
194      SET COLOR TO W/B, W/B
195      @ 24,0 SAY SPACE(80)
196      SET COLOR TO W+/R, W+/R
197      @ 24,16 SAY " Day portion of date must be between 01 and 30 "
198      DO DELAY
199      SET COLOR TO /W, /W
200      @ 24,0 SAY MESSAGE
```

```
201      STORE SUBSTR(MDATE,5,2) TO MDAY
202      SET COLOR TO R+/B, R+B
203      @ 3,49 GET MDAY PICT "99"
204      READ
205      STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
206      LOOP
207  ELSE
208  *
209      IF (SUBSTR(MOLDATE,3,2) = "02" .AND. .NOT.;
210         (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
211         SUBSTR(MOLDATE,5,2) <= "28")) THEN
212          SET COLOR TO W/B, W/B
213          @ 24,0 SAY SPACE(80)
214          SET COLOR TO W+/R, W+/R
215          @ 24,16 SAY " Day portion of date must be between 01 and 28 "
216          DO DELAY
217          SET COLOR TO /W, /W
218          @ 24,0 SAY MESSAGE
219          STORE SUBSTR(MDATE,5,2) TO MDAY
220          SET COLOR TO R+/B, R+B
221          @ 3,49 GET MDAY PICT "99"
222          READ
223          STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
224          LOOP
225  ELSE
226  *
227      IF .NOT. (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
228              SUBSTR(MOLDATE,5,2) <= "31") THEN
229          SET COLOR TO W/B, W/B
230          @ 24,0 SAY SPACE(80)
231          SET COLOR TO W+/R, W+/R
232          @ 24,16 SAY " Day portion of date must be between 01 and 31 "
233          DO DELAY
234          SET COLOR TO /W, /W
235          @ 24,0 SAY MESSAGE
236          STORE SUBSTR(MDATE,5,2) TO MDAY
237          SET COLOR TO R+/B, R+B
238          @ 3,49 GET MDAY PICT "99"
239          READ
240          STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
241          LOOP
242  ELSE
243      EXIT
244  ENDIF
245  ENDIF
246  ENDIF
247  ENDDO WHILE .T.
248  *
249  GO TOP
250  FIND &MOLDATE
```

```
251 IF EOF() = .T. THEN
252     SET COLOR TO W/B, W/B
253     @ 24,0 SAY SPACE(80)
254     STORE "EFFECTIVE DATE " + MOLDATE + " does not exist for site " +
255         MSITE + ", try another " TO NODATE
256     SET COLOR TO W+/R, W+/R
257     @ 24,06 SAY NODATE
258     DO DELAY
259     SET COLOR TO /W, /W
260     @ 24,0 SAY MESSAGE
261     STORE "000000" TO MOLDATE
262     LOOP
263 ELSE
264     EXIT
265 ENDIF EOF() = .T.
266 ENDDO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDATE)
267 *
268 SET COLOR TO W+/B, W+/B
269 @ 05,05 SAY SPACE(70)
270 @ 24,0 SAY SPACE(80)
271 *
272 * CLEAR LISTING OF EFFECTIVE DATES FROM SCREEN
273 *
274 SET COLOR TO /BR, /BR
275 @ 07,2 SAY SPACE(76)
276 @ 08,2 SAY SPACE(76)
277 @ 09,2 SAY SPACE(76)
278 @ 10,2 SAY SPACE(76)
279 @ 11,2 SAY SPACE(76)
280 @ 12,2 SAY SPACE(76)
281 @ 13,2 SAY SPACE(76)
282 @ 14,2 SAY SPACE(76)
283 @ 15,2 SAY SPACE(76)
284 @ 16,2 SAY SPACE(76)
285 @ 17,2 SAY SPACE(76)
286 @ 18,2 SAY SPACE(76)
287 @ 19,2 SAY SPACE(76)
288 @ 20,2 SAY SPACE(76)
289 @ 21,2 SAY SPACE(76)
290 *
291 SET COLOR TO R+/ , R+/
292 @ 13,18 SAY " CREATING TEMPORARY DATABASE AND INDEX FILE "
293 STORE "MOLDATE" + "MSITE" TO MKEY
294 GO TOP
295 FIND &MKEY
296 *
297 COPY TO TEMPONE FOR SITENO = "&MSITE" .AND. EFFDATE = "&MOLDATE"
298 SELECT 1
299 USE TEMPONE
300 INDEX ON FEATURENO TO TEMPONE
```

```
301 SELECT 2
302 USE DESCRIP INDEX DESCRIP
303 SELECT TEMPONE
304 SET RELATION TO FEATURENO INTO DESCRIP
305 GO TOP
306 *
307 *   CREATE THE SPLICE EQUIPMENT PROJECT REPORT AND CHECK IF THE REPORT
308 *   IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
309 *
310 SET COLOR TO W+/BR, W+/BR
311 @ 13,15 SAY SPACE(60)
312 @ 13,16 SAY " Do you want a printed report? (Yes or No): "
313 SET COLOR TO /BR, /BR
314 @ 13,49 SAY "Y"
315 @ 13,56 SAY "N"
316 STORE "N" TO ACCEPT
317 @ 13,62 GET ACCEPT PICT "!"
318 READ
319 *
320 *   ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
321 *
322 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
323     IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
324         SET COLOR TO W+/R, W+/R
325         @ 24,24 SAY " Response must be either N or Y "
326         DO DELAY
327         STORE "N" TO ACCEPT
328     ENDIF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
329     SET COLOR TO /BR, /BR
330     @ 13,62 GET ACCEPT PICT "!"
331     READ
332 ENDDO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
333 *
334 SET COLOR TO /BR, /BR
335 @ 13,15 SAY SPACE(55)
336 *
337 IF ACCEPT = "Y" THEN
338     ?? FLASH + "W.PRINTER/"
339     SET CONSOLE OFF
340     WAIT TO CHOICE
341     SET CONSOLE ON
342     SET COLOR TO W/B, W/B
343     @ 22,10 SAY SPACE(65)
344     STORE DTOC(DATE()) TO TODAY
345     STORE SUBSTR(TODAY,4,2) + " " + CMONTH(DATE()) + " 19" +;
346         SUBSTR(TODAY,7,2) TO TODATE
347     STORE 0 TO PAGENO
348     STORE 61 TO LINECT
349     SET COLOR TO R+/ , R+/
350     SET DEVICE TO PRINT
```

```

351  *
352  DO WHILE .NOT. EOF()
353      DO WHILE (LINECT <= 60 .AND. .NOT. EOF())
354          @ LINECT,3 SAY SITENO PICT "99"
355          @ LINECT,9 SAY B->CLIN PICT "9999"
356          @ LINECT,17 SAY FEATURENO PICT "999999"
357          @ LINECT,28 SAY B->DESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
358          @ LINECT,60 SAY QTY PICT "999"
359          @ LINECT,66 SAY UNIT PRICE PICT "99999999.99"
360          LINECT = LINECT + 1
361          SKIP
362      ENDDO WHILE (LINECT <= 60 .AND. .NOT. EOF())
363  *
364  IF EOF() = .T. THEN
365      IF PAGENO > 1 THEN
366          @ 62,37 SAY "Page " + STR(PAGENO,2,0)
367      ENDIF PAGENO > 1
368      EJECT
369      SET DEVICE TO SCREEN
370      @ 13,25 SAY " FINISHED PRINTING THE REPORT "
371      DO DELAY
372      EXIT
373  ELSE
374      SET DEVICE TO SCREEN
375      @ 13,27 SAY " Printing Page Number " + STR(PAGENO + 1,2,0) + " "
376      SET DEVICE TO PRINT
377  ENDIF EOF() = .T.
378  *
379  IF (LINECT > 60 .AND. PAGENO > 1) THEN
380      @ 62,37 SAY "Page " + STR(PAGENO,2,0)
381  ENDIF (LINECT > 60 .AND. PAGENO > 1)
382  @ 2,25 SAY " DELIVERY ORDER LEVEL REPORT "
383  @ 3,29 SAY "EFFECTIVE DATE: "
384  @ 3,45 SAY MOLDATE
385  @ 4,60 SAY TODATE
386  @ 6,2 SAY "SITE CLIN. FEATURE# DESCRIPTION"
387  @ 6,60 SAY "QTY UNIT PRICE "
388  @ 7,2 SAY "===== "
389  @ 7,51 SAY "===== "
390  PAGENO = PAGENO + 1
391  STORE 9 TO LINECT
392  *
393  ENDDO WHILE .NOT. EOF()
394  ELSE
395      SET COLOR TO GR+/B, GR+/B
396      @ 5,2 SAY "SITE CLIN. FEATURE# DESCRIPTION"
397      @ 5,60 SAY "QTY UNIT PRICE "
398      SET COLOR TO /BR, /BR
399      STORE 0 TO LINECT
400  *

```

```
401 DO WHILE .NOT. EOF()
402   DO WHILE LINECT < 15
403     @ LINECT+7,3 SAY SITENO PICT "99"
404     @ LINECT+7,9 SAY B->CLIN PICT "9999"
405     @ LINECT+7,17 SAY FEATURENO PICT "999999"
406     @ LINECT+7,28 SAY B->DESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
407     @ LINECT+7,60 SAY QTY PICT "999"
408     @ LINECT+7,66 SAY UNIT_PRICE PICT "99999999.99"
409     LINECT = LINECT + 1
410     SKIP
411     IF EOF() = .T. THEN
412       SET COLOR TO W+/R, W+/R
413       @ 24,18 SAY " End of File reached, Press any key to EXIT "
414       SET CONSOLE OFF
415       WAIT TO ACCEPT
416       SET CONSOLE ON
417       EXIT
418     ENDIF EOF() = .T.
419   ENDDO WHILE LINECT < 15
420 *
421   IF EOF() = .T. THEN
422     EXIT
423   ENDIF EOF() = .T.
424   SET COLOR TO R+/B, R+/B
425   STORE "C" TO CHOICE
426   @ 22,57 GET CHOICE PICT "!"
427   READ
428 *
429 *   ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
430 *
431   DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
432     IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
433       SET COLOR TO W+/R, W+/R
434       @ 24,24 SAY " Response must be either C or X "
435       DO DELAY
436       STORE "C" TO CHOICE
437     ENDIF .NOT. (CHOICE = "C" .OR. CHOICE = "X")
438     SET COLOR TO R+/B, R+/B
439     @ 22,57 GET CHOICE PICT "!"
440     READ
441   ENDDO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
442 *
443 *   DETERMINE IF THE USER WANTS TO QUIT OR CONTINUE
444 *
445   IF CHOICE = "C"
446     SET COLOR TO /BR, /BR
447     @ 07,2 SAY SPACE(76)
448     @ 08,2 SAY SPACE(76)
449     @ 09,2 SAY SPACE(76)
450     @ 10,2 SAY SPACE(76)
```


Page 10

EQPDTPRC.PRG Program Listing

```
451      @ 11,2 SAY SPACE(76)
452      @ 12,2 SAY SPACE(76)
453      @ 13,2 SAY SPACE(76)
454      @ 14,2 SAY SPACE(76)
455      @ 15,2 SAY SPACE(76)
456      @ 16,2 SAY SPACE(76)
457      @ 17,2 SAY SPACE(76)
458      @ 18,2 SAY SPACE(76)
459      @ 19,2 SAY SPACE(76)
460      @ 20,2 SAY SPACE(76)
461      @ 21,2 SAY SPACE(76)
462      STORE 0 TO LINECT
463      ELSE
464      EXIT
465      ENDIF CHOICE = "C"
466      *
467      ENDDO WHILE .NOT. EOF()
468      *
469      ENDIF ACCEPT = "Y"
470      *
471      * ERASE ALL TEMPORARY FILES CREATED DURING REPORT GENERATION
472      *
473      CLOSE DATABASES
474      SET CONSOLE OFF
475      ERASE TEMPONE.DBF
476      ERASE TEMPONE.NDX
477      SET CONSOLE ON
478      SET PRINT OFF
479      *
480      * RETURN TO CALLING PROGRAM
481      *
482      RELEASE ALL LIKE M*, ACCEPT, CHOICE, COLCNT, ERROR, LINECT, PAGENO,;
483      SYSDATE, TODAY, TODATE
484      RETURN
485      *****
```

Page 1

EQPPJRPT.PRG Program Listing

```

1  * PROCEDURE EQPPJRPT.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER A SPLICE EQUIPMENT
9  *               PROJECT LEVEL REPORT.
10 *
11 * INPUT FILES   : EQUIP.DBF, DESCRIP.DBF, DECSRIP.NDX,
12 *               TEMPONE.DBF, EFEAT.NDX
13 *
14 * OUTPUT FILE   : TEMPONE.DBF
15 *
16 * CALLED BY     : PROJRPPTS.PRG
17 *
18 * MODULES CALLED : DELAY.PRG
19 *
20 * LOCAL VARIABLES: ACCEPT, CHOICE, LINECT, PAGENO, TODAY, TODATE
21 *
22 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
23 *
24 * CASE SELECTION = 1   EQUIPMENT PROJECT LEVEL REPORT
25 *
26 * CALL EQUIPMENT DATABASE INDEXED ON CONTRACT LINE NUMBER AND FEATURE
27 * NUMBER AND TOTAL ON QUANTITY. RELATE TO DESCRIP FILE ON FEATURENO.
28 *
29 SET ESCAPE OFF
30 SET TALK OFF
31 SET COLOR TO W+/B, W+/B, B
32 CLEAR
33 USE EQUIP
34 GO TOP
35 IF EOF() = .T. THEN
36     SET COLOR TO W+/R, W+/R
37     @ 13,24 SAY " The EQUIPMENT Database is EMPTY! "
38     DO DELAY
39     RETURN
40 ENDIF
41 ?? FLASH + "S.REPORTS.SCR/"
42 @ 24,0 SAY SPACE(80)
43 SET COLOR TO R+/ , R+/
44 @ 2,25 SAY " EQUIPMENT PROJECT LEVEL REPORT "
45 *
46 * CREATE THE SPLICE EQUIPMENT PROJECT REPORT AND CHECK IF THE REPORT
47 * IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
48 *
49 SET COLOR TO W+/BR, W+/BR
50 @ 13,16 SAY " Do you want a printed report? (Yes or No): "

```

AD-A168 517

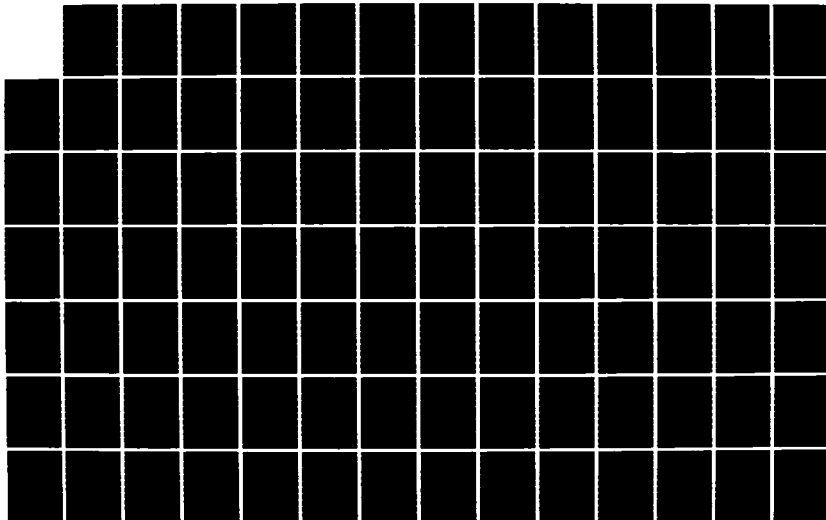
DEVELOPMENT OF AN AUTOMATED MICROCOMPUTER
KNOWLEDGE-BASED INTEGRATED CON. (U) NAVAL POSTGRADUATE
SCHOOL MONTEREY CA R L BEARD MAR 86

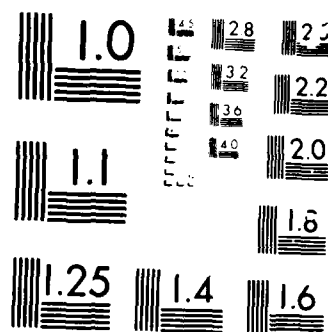
4/5

UNCLASSIFIED

F/G 9/2

NL





W. J. P. 1.1

Page 2

EQPPJRPT.PRG Program Listing

```
51 SET COLOR TO /BR, /BR
52 @ 13,49 SAY "Y"
53 @ 13,56 SAY "N"
54 STORE "N" TO ACCEPT
55 @ 13,62 GET ACCEPT PICT "!"
56 READ
57 *
58 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
59 *
60 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
61 IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
62 SET COLOR TO W+/R, W+/R
63 @ 24,24 SAY " Response must be either N or Y "
64 DO DELAY
65 STORE "N" TO ACCEPT
66 ENDIF
67 SET COLOR TO /BR, /BR
68 @ 13,62 GET ACCEPT PICT "!"
69 READ
70 ENDDO
71 *
72 SET COLOR TO /BR, /BR
73 @ 13,15 SAY SPACE(55)
74 *
75 SET COLOR TO W+/BR, W+/BR
76 @ 13,19 SAY " COMPUTING TOTALS FOR EACH FEATURE NUMBER "
77 *
78 USE EQUIP INDEX EFPEAT
79 GO TOP
80 SET CONSOLE OFF
81 ERASE TEMPONE.DBF
82 SET CONSOLE ON
83 *
84 * COMPUTE THE TOTAL QUANTITY FOR EACH FEATURE NUMBER
85 *
86 TOTAL ON FEATURENO TO TEMPONE.DBF FIELDS QTY WHILE FEATURENO <> 'XXXXXX'
87 *
88 SELECT 1
89 USE TEMPONE
90 SELECT 2
91 USE DESCRIP INDEX DESCRIP
92 SELECT TEMPONE
93 SET RELATION TO FEATURENO INTO DESCRIP
94 GO TOP
95 *
96 @ 13,15 SAY SPACE(55)
97 *
98 IF ACCEPT = "Y" THEN
99 ?? FLASH + "W.PRINTER/"
100 SET CONSOLE OFF
```

```

101 WAIT TO CHOICE
102 SET CONSOLE ON
103 SET COLOR TO W/B, W/B
104 @ 22,10 SAY SPACE(65)
105 STORE DIOC( DATE() ) TO TODAY
106 STORE SUBSTR( TODAY, 4, 2 ) + " " + CMONTH( DATE() ) + " 19" + ;
107     SUBSTR( TODAY, 7, 2 ) TO TODATE
108 STORE 0 TO PAGENO
109 STORE 61 TO LINECT
110 SET COLOR TO R+ / , R+ /
111 SET DEVICE TO PRINT
112 *
113 DO WHILE .NOT. EOF()
114     DO WHILE ( LINECT <= 60 .AND. .NOT. EOF() )
115         @ LINECT, 10 SAY DESCRIP->CLIN
116         @ LINECT, 22 SAY FEATURENO
117         @ LINECT, 35 SAY DESCRIP->DESCRIPT
118         @ LINECT, 68 SAY QTY
119         LINECT = LINECT + 1
120         SKIP
121     ENDDO WHILE
122 *
123 IF EOF() = .T. THEN
124     IF PAGENO > 1 THEN
125         @ 62, 37 SAY "Page " + STR( PAGENO, 2, 0 )
126     ENDIF
127     EJECT
128     SET DEVICE TO SCREEN
129     @ 13, 25 SAY " FINISHED PRINTING THE REPORT "
130     DO DELAY
131     EXIT
132 ELSE
133     SET DEVICE TO SCREEN
134     @ 13, 27 SAY " Printing Page Number " + STR( PAGENO + 1, 2, 0 ) + " "
135     SET DEVICE TO PRINT
136 ENDIF
137 *
138 IF ( LINECT > 60 .AND. PAGENO > 1 ) THEN
139     @ 62, 37 SAY "Page " + STR( PAGENO, 2, 0 )
140 ENDIF
141 @ 2, 25 SAY " EQUIPMENT PROJECT LEVEL REPORT "
142 @ 4, 60 SAY TODATE
143 @ 6, 10 SAY "CLIN          FEATURE#          DESCRIPTION"
144 @ 6, 68 SAY "QTY"
145 @ 7, 2 SAY "===== "
146 @ 7, 51 SAY "===== "
147 PAGENO = PAGENO + 1
148 STORE 9 TO LINECT
149 *
150 ENDDO WHILE .NOT. EOF()

```

Page 4

EQPPJRPT.PRG Program Listing

```
151 *
152 ELSE
153     SET COLOR TO GR+/B, GR+/B
154     @ 5,10 SAY "CLIN      FEATURE#      DESCRIPTION"
155     @ 5,68 SAY "QTY"
156     SET COLOR TO /BR, /BR
157     STORE 0 TO LINECT
158 *
159     DO WHILE .NOT. EOF()
160         DO WHILE LINECT < 15
161             @ LINECT+7,10 SAY DESCRIP->CLIN
162             @ LINECT+7,22 SAY FEATURENO
163             @ LINECT+7,35 SAY DESCRIP->DESCRIPT
164             @ LINECT+7,68 SAY QTY
165             LINECT = LINECT + 1
166             SKIP
167             IF EOF() = .T. THEN
168                 SET COLOR TO W+/R, W+/R
169                 @ 24,18 SAY " End of File reached, Press any key to EXIT "
170                 SET CONSOLE OFF
171                 WAIT TO ACCEPT
172                 SET CONSOLE ON
173                 EXIT
174             ENDIF
175         ENDDO WHILE LINECT < 15
176 *
177         IF EOF() = .T. THEN
178             EXIT
179         ENDIF
180         SET COLOR TO R+/B, R+/B
181         STORE "C" TO CHOICE
182         @ 22,57 GET CHOICE PICT "!"
183         READ
184 *
185 *         ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
186 *
187         DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
188             IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
189                 SET COLOR TO W+/R, W+/R
190                 @ 24,24 SAY " Response must be either C or X "
191                 DO DELAY
192                 STORE "C" TO CHOICE
193             ENDIF
194             SET COLOR TO R+/B, R+/B
195             @ 22,57 GET CHOICE PICT "!"
196             READ
197         ENDDO
198 *
199 *         DETERMINE IF THE USER WANTS TO QUIT OR CONTINUE
200 *
```

```
201      IF CHOICE = "C"
202          SET COLOR TO /BR, /BR
203          @ 07,2 SAY SPACE(76)
204          @ 08,2 SAY SPACE(76)
205          @ 09,2 SAY SPACE(76)
206          @ 10,2 SAY SPACE(76)
207          @ 11,2 SAY SPACE(76)
208          @ 12,2 SAY SPACE(76)
209          @ 13,2 SAY SPACE(76)
210          @ 14,2 SAY SPACE(76)
211          @ 15,2 SAY SPACE(76)
212          @ 16,2 SAY SPACE(76)
213          @ 17,2 SAY SPACE(76)
214          @ 18,2 SAY SPACE(76)
215          @ 19,2 SAY SPACE(76)
216          @ 20,2 SAY SPACE(76)
217          @ 21,2 SAY SPACE(76)
218          STORE 0 TO LINECT
219      ELSE
220          EXIT
221      ENDIF
222      *
223      ENDDO WHILE .NOT. EOF()
224      *
225  ENDIF
226      *
227      * ERASE THE TEMPORARY DATABASE USED FOR TOTALS
228      *
229      CLOSE DATABASES
230      SET CONSOLE OFF
231      ERASE TEMPONE.DBF
232      SET CONSOLE ON
233      SET PRINT OFF
234      *
235      * RETURN TO CALLING PROGRAM
236      *
237      RELEASE ACCEPT, CHOICE, LINECT, PAGENO, TODAY, TDATE
238      RETURN
239      *****
```


Page 1

EQPSTRPT.PRG Program Listing

```

1  * PROCEDURE EQPSTRPT.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER A SPLICE EQUIPMENT SITE
9  *               LEVEL REPORT FOR A SINGLE SITE.
10 *
11 * INPUT FILES   : EQUIP.DBF, EFEAT.NDX, DESCRIP.DBF, DESCRIP.NDX,
12 *               TEMPONE.DBF, EQUIPSIT.NDX
13 *
14 * OUTPUT FILES  : NONE.
15 *
16 * CALLED BY     : SITERPTS.PRG
17 *
18 * MODULES CALLED : DELAY.PRG
19 *
20 * GLOBAL VARIABLE: HISITE, LOSITE
21 *
22 * LOCAL VARIABLES: ACCEPT, CHOICE, ERROR, LINECT, MESSAGE, MSITE,
23 *               PAGENO, TODAY, TODATE
24 *
25 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
26 *
27 * CASE SELECTION = 1      EQUIPMENT SITE LEVEL REPORT
28 *
29 * CALL EQUIPMENT DATABASE INDEXED ON SITE NUMBER, CONTRACT LINE NUMBER
30 * AND FEATURE NUMBER AND TOTAL ON QUANTITY.
31 *
32 SET ESCAPE OFF
33 SET TALK OFF
34 SET COLOR TO W+/B, W+/B, B
35 CLEAR
36 USE EQUIP
37 GO TOP
38 IF EOF() = .T. THEN
39     SET COLOR TO W+/R, W+/R
40     @ 13,24 SAY " The EQUIPMENT Database is EMPTY! "
41     DO DELAY
42     RETURN
43 ENDIF
44 ?? FLASH + "S.REPORTS.SCR/"
45 @ 24,0 SAY SPACE(80)
46 SET COLOR TO R+/ , R+/
47 @ 2,26 SAY " EQUIPMENT SITE LEVEL REPORT "
48 *
49 * ENSURE THAT TEMPORARY DATABASE DOES NOT EXIST, IF SO ERASE IT
50 *

```

Page 2

EQPSTRPT.PRG Program Listing

```

51 SET CONSOLE OFF
52 ERASE TEMPONE.DBF
53 SET CONSOLE ON
54 *
55 SET COLOR TO W+/BR, W+/BR
56 @ 13,15 SAY "Enter site number for which the report is desired:"
57 *
58 DO WHILE .T.
59     SET COLOR TO /BR, /BR
60     STORE LOSITE TO MSITE
61     @ 13,66 GET MSITE PICT '99'
62     READ
63     IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
64         SET COLOR TO W+/R, W+/R
65         STORE ' Response must be between ' + LOSITE + ;
66             'and ' + HISITE + ' ' TO ERROR
67         @ 24,22 SAY ERROR
68         DO DELAY
69         LOOP
70     ELSE
71         USE EQUIP INDEX EQUIPSIT
72         GO TOP
73         FIND &MSITE
74         IF EOF() = .T. THEN
75             STORE " No equipment exists for site " + MSITE + ;
76                 ", try another site " TO MESSAGE
77             SET COLOR TO W+/R, W+/R
78             @ 24,15 SAY MESSAGE
79             DO DELAY
80             LOOP
81         ELSE
82             EXIT
83         ENDIF EOF() = .T.
84     ENDIF
85 ENDDO WHILE .T.
86 *
87 SET COLOR TO W+/BR, W+/BR
88 @ 13,15 SAY SPACE(55)
89 *
90 *   CREATE THE SPLICE EQUIPMENT PROJECT REPORT AND CHECK IF THE REPORT
91 *   IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
92 *
93 @ 13,16 SAY " Do you want a printed report? (Yes or No): "
94 SET COLOR TO /BR, /BR
95 @ 13,49 SAY "Y"
96 @ 13,56 SAY "N"
97 STORE "N" TO ACCEPT
98 @ 13,62 GET ACCEPT PICT "!"
99 READ
100 *

```

```

101 *   ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
102 *
103 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
104     IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
105         SET COLOR TO W+/R, W+/R
106         @ 24,24 SAY " Response must be either N or Y "
107         DO DELAY
108         STORE "N" TO ACCEPT
109     ENDIF
110     SET COLOR TO /BR, /BR
111     @ 13,62 GET ACCEPT PICT "!"
112     READ
113 ENDDO
114 SET COLOR TO /BR, /BR
115 @ 13,15 SAY SPACE(55)
116 *
117 SET COLOR TO W+/BR, W+/BR
118 @ 13,17 SAY " COMPUTING TOTALS FOR EACH SITE FEATURE NUMBER "
119 *
120 USE EQUIP INDEX EFEAT
121 TOTAL ON FEATURENO TO TEMPONE.DBF FIELDS QTY;
122     FOR FEATURENO <> 'XXXXXX' .AND. SITENO = '&MSITE'
123 SELECT 1
124 USE TEMPONE
125 SELECT 2
126 USE DESCRIP INDEX DESCRIP
127 SELECT TEMPONE
128 SET RELATION TO FEATURENO INTO DESCRIP
129 GO TOP
130 *
131 @ 13,15 SAY SPACE(55)
132 *
133 IF ACCEPT = "Y" THEN
134     ?? FLASH + "W.PRINTER/"
135     SET CONSOLE OFF
136     WAIT TO CHOICE
137     SET CONSOLE ON
138     SET COLOR TO W/B, W/B
139     @ 22,10 SAY SPACE(65)
140     STORE DTOC(DATE()) TO TODAY
141     STORE SUBSTR(TODAY,4,2) + " " + CMONTH(DATE()) + " 19" + ;
142         SUBSTR(TODAY,7,2) TO TODATE
143     STORE 0 TO PAGENO
144     STORE 61 TO LINECT
145     SET COLOR TO R+/ , R+/
146     SET DEVICE TO PRINT
147 *
148 DO WHILE .NOT. EOF()
149     DO WHILE (LINECT <= 60 .AND. .NOT. EOF())
150         @ LINECT,9 SAY SITENO

```

Page 4

EQPS'IRPT.PRG Program Listing

```

151      @ LINECT,17 SAY DESCRIP->CLIN
152      @ LINECT,27 SAY FEATURENO
153      @ LINECT,39 SAY DESCRIP->DESCRIPT
154      @ LINECT,71 SAY QTY
155      LINECT = LINECT + 1
156      SKIP
157      ENDDO WHILE (LINECT <= 60 .AND. .NOT. EOF())
158  *
159      IF EOF() = .T. THEN
160          IF PAGENO > 1 THEN
161              @ 62,37 SAY "Page " + STR(PAGENO,2,0)
162          ENDIF PAGENO > 1
163          EJECT
164          SET DEVICE TO SCREEN -
165          @ 13,25 SAY " FINISHED PRINTING THE REPORT "
166          DO DELAY
167          EXIT
168      ELSE
169          SET DEVICE TO SCREEN
170          @ 13,27 SAY " Printing Page Number " + STR(PAGENO + 1,2,0) + " "
171          SET DEVICE TO PRINT
172      ENDIF EOF() = .T.
173  *
174      IF (LINECT > 60 .AND. PAGENO > 1) THEN
175          @ 62,37 SAY "Page " + STR(PAGENO,2,0)
176      ENDIF (LINECT > 60 .AND. PAGENO > 1)
177      @ 2,25 SAY " EQUIPMENT SITE LEVEL REPORT "
178      @ 4,60 SAY TODAYE
179      @ 6,8 SAY "SITE      CLIN      FEATURE#      DESCRIPTION"
180      @ 6,71 SAY "QTY"
181      @ 7,2 SAY "===== "
182      @ 7,51 SAY "===== "
183      PAGENO = PAGENO + 1
184      STORE 9 TO LINECT
185  *
186      ENDDO WHILE .NOT. EOF()
187  *
188      ELSE
189          SET COLOR TO GR+/B, GR+/B
190          @ 5,8 SAY "SITE      CLIN      FEATURE#      DESCRIPTION"
191          @ 5,71 SAY "QTY"
192          SET COLOR TO /BR, /BR
193          STORE 0 TO LINECT
194  *
195      DO WHILE .NOT. EOF()
196          DO WHILE LINECT < 15
197              @ LINECT+7,9 SAY SITE#
198              @ LINECT+7,17 SAY DESCRIP->CLIN
199              @ LINECT+7,27 SAY FEATURENO
200              @ LINECT+7,39 SAY DESCRIP->DESCRIPT

```

```
201 @ LINECT+7,71 SAY QTY
202 LINECT = LINECT + 1
203 SKIP
204 IF EOF() = .T. THEN
205     SET COLOR TO W+/R, W+/R
206     @ 24,18 SAY " End of File reached, Press any key to EXIT "
207     SET CONSOLE OFF
208     WAIT TO ACCEPT
209     SET CONSOLE ON
210     EXIT
211 ENDIF EOF() = .T.
212 ENDDO WHILE LINECT < 15
213 *
214 IF EOF() = .T..THEN
215     EXIT
216 ENDIF EOF() = .T.
217 SET COLOR TO R+/B, R+/B
218 STORE "C" TO CHOICE
219 @ 22,57 GET CHOICE PICT "!"
220 READ
221 *
222 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
223 *
224 DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
225     IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
226         SET COLOR TO W+/R, W+/R
227         @ 24,24 SAY " Response must be either C or X "
228         DO DELAY
229         STORE "C" TO CHOICE
230     ENDIF .NOT. (CHOICE = "C" .OR. CHOICE = "X")
231     SET COLOR TO R+/B, R+/B
232     @ 22,57 GET CHOICE PICT "!"
233     READ
234 ENDDO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
235 *
236 * DETERMINE IF THE USER WANTS TO QUIT OR CONTINUE
237 *
238 IF CHOICE = "C"
239     SET COLOR TO /BR, /BR
240     @ 07,2 SAY SPACE(76)
241     @ 08,2 SAY SPACE(76)
242     @ 09,2 SAY SPACE(76)
243     @ 10,2 SAY SPACE(76)
244     @ 11,2 SAY SPACE(76)
245     @ 12,2 SAY SPACE(76)
246     @ 13,2 SAY SPACE(76)
247     @ 14,2 SAY SPACE(76)
248     @ 15,2 SAY SPACE(76)
249     @ 16,2 SAY SPACE(76)
250     @ 17,2 SAY SPACE(76)
```

```

251      @ 18,2 SAY SPACE(76)
252      @ 19,2 SAY SPACE(76)
253      @ 20,2 SAY SPACE(76)
254      @ 21,2 SAY SPACE(76)
255      STORE 0 TO LINECT
256      ELSE
257          EXIT
258      ENDIF CHOICE = "C"
259      *
260      ENDDO WHILE .NOT. EOF()
261      *
262      ENDIF ACCEPT = "Y"
263      *
264      *   ERASE THE TEMPORARY DATABASE USED FOR TOTALS
265      *
266      CLOSE DATABASES
267      SET CONSOLE OFF
268      ERASE TEMPONE.DBF
269      SET CONSOLE ON
270      SET PRINT OFF
271      *
272      *   RETURN TO CALLING PROGRAM
273      *
274      RELEASE ACCEPT, CHOICE, ERROR, LINECT, MESSAGE, MSITE, PAGENO,;
275          TODAY, TDATE
276      RETURN
277      *****

```

```

1  * PROCEDURE EQUIPCMD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              : LCDR WINSTON H. BUCKLEY, SC, USN
5  *              : LCDR ROBERT F. BRADO, USN
6  *              : LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER THE OPPORTUNITY TO MODIFY OR REVIEW
9  *              : ALL DATA IN THE EQUIPMENT DATABASE.
10 *
11 * INPUT FILES   : NONE
12 *
13 * OUTPUT FILE   : NONE
14 *
15 * MODULES CALLED : EQUIPUPD.PRG. EQUIPREV.PRG
16 *
17 * CALLED BY     : MAINMENU.CMD
18 *
19 * LOCAL VARIABLES: SELEKT
20 *
21 * DATE LAST TIME MODIFIED =====> 22 DECEMBER 1985 <=====
22 *
23 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE
24 * SELECTION.
25 *
26 STORE "1" TO SELEKT
27 DO WHILE SELEKT = "1"
28   SET OVERWRITE TO OFF, WAIT
29   CLEAR
30   ?? FLASH * "WELCOME "
31   SET OVERWRITE ON
32   WAIT TO THE LEFT
33   SET OVERWRITE ON
34 *
35 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
36 *
37 DO CASE
38 *
39 *   CALL THE EQUIPMENT UPDATE PROGRAM.
40   CASE SELEKT = "1"
41     DO EQUIPUPD
42 *
43 *   CALL THE EQUIPMENT REVIEW PROGRAM.
44   CASE SELEKT = "2"
45     DO EQUIPREV
46 *
47 *   RETURN TO THE MAIN MENU PROGRAM.
48   CASE SELEKT = "3"
49 *
50   ENDCASE

```

Page 2

EQUIPCMD.PRG Program Listing

```
51 | *
52 | ENDDO (WHILE SELEKT = "3")
53 | *
54 | * RETURN TO THE CALLING PROGRAM
55 | *
56 | RETURN
57 | *****
```



```

1 * PROCEDURE EQUIPREV.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. BUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, SC, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : TO ENABLE THE USER TO REVIEW ALL RECORDS IN THE
9 *              : EQUIPMENT DATABASE.
10 *
11 * INPUT FILES  : EQUIP.DBF INDEX EQUIPSIT.NDX
12 *
13 * OUTPUT FILES : NONE
14 *
15 * CALLED BY    : EQUIPCMD.PRG
16 *
17 * MODULES CALLED : DELAY.PRG
18 *
19 * GLOBAL VARIABLE: HDATE, HIFNUM, HISITE, LDATE, LOFNUM, LOSITE
20 *
21 * LOCAL VARIABLES: CURRENTNO, EOF, ERROR, FIRST_REC, LAST_REC, MCLIN,
22 *                 MDESCPT, MESSAGE, MSITE, MFENT, TOF
23 *
24 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
25 *
26 * CASE SELECTION = 2      REVIEW EQUIPMENT FILE RECORDS
27 *
28 * USE EQUIPMENT DATABASE INDEXED ON SITE NUMBER AND WAIT FOR THE
29 * USER TO INPUT THE SITE NUMBER, THEN START REVIEWING FROM THAT POINT.
30 *
31 SET ESCAPE OFF
32 SET TALK OFF
33 SELECT 1
34 USE EQUIP
35 GO TOP
36 SET COLOR TO W+/B, W+/B, B
37 CLEAR
38 IF EOF() = .T. THEN
39     SET COLOR TO W+/R, W+/R
40     @ 13,24 SAY " The EQUIPMENT Database is EMPTY! "
41     DO DELAY
42     RETURN
43 ENDIF
44 ?? FLASH + "S.EQUIPREV.SCR/"
45 @ 24,0 SAY SPACE (80)
46 STORE "Enter 00 to start at EOF, 99 to start at EOF, or a site number " +;
47     "between 01 and 58 " TO MESSAGE
48 SET COLOR TO /W, /W
49 @ 24,0 SAY MESSAGE
50 STORE '88' TO MSITE

```

Page 2

EQUIPREV.PRG Program Listing

```

51 DO WHILE .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
52   SET COLOR TO /BR, /BR
53   STORE '00' TO MSITE
54   @ 9,20 GET MSITE PICT '99'
55   READ
56   IF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
57     SET COLOR TO W/B, W/B
58     @ 24,0 SAY SPACE(80)
59     SET COLOR TO W+/R, W+/R
60     STORE ' Response must be between ' + LOSITE + ' and ' +;
61       HISITE + ', Zero (00) or 99 ' TO ERROR
62     @ 24,13 SAY ERROR
63     DO DELAY
64     SET COLOR TO /W, /W
65     @ 24,0 SAY MESSAGE
66     LOOP
67   ELSE
68     IF (MSITE = '00' .OR. MSITE = '99') THEN
69       USE EQUIP
70       IF MSITE = '00' THEN
71         GO BOTTOM
72         STORE RECNO() TO LAST_REC
73         GO TOP
74         STORE RECNO() TO FIRST_REC
75       ELSE
76         IF MSITE = '99' THEN
77           GO TOP
78           STORE RECNO() TO FIRST_REC
79           GO BOTTOM
80           STORE RECNO() TO LAST_REC
81         ENDIF MSITE = '99'
82       ENDIF MSITE = '00'
83     ELSE
84       USE EQUIP INDEX EQUIPSIT, EQUIPPRJ, EQUIPDAT, EQUIPSD
85       GO TOP
86       FIND &MSITE
87       IF EOF() = .T. THEN
88         SET COLOR TO W/B, W/B
89         @ 24,0 SAY SPACE(80)
90         STORE " No records exist for site number " + MSITE +;
91           ", try again " TO ERROR
92         SET COLOR TO W+/R, W+/R
93         @ 24,16 SAY ERROR
94         DO DELAY
95         SET COLOR TO /W, /W
96         @ 24,0 SAY MESSAGE
97         STORE '88' TO MSITE
98       ENDIF
99     ENDIF
100  ENDIF

```

```

101 ENDDO WHILE
102 *
103 STORE SPACE(10) + 'Enter "00      " to start at TOF or a six digit ' +;
104     'feature number' + SPACE(10) TO MESSAGE
105 IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
106     SET COLOR TO /W, /W
107     @ 24,0 SAY MESSAGE
108     DO WHILE .T.
109         SET COLOR TO /BR, /BR
110         STORE '00      ' TO MFEAT
111         @ 14,45 GET MFEAT PICT '999999'
112         READ
113         IF .NOT. ((MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) .OR.;
114             MFEAT = '00      ' .OR. MFEAT = '99      ')
115             SET COLOR TO W/B, W/B
116             @ 24,0 SAY SPACE(80)
117             SET COLOR TO W+/R, W+/R
118             STORE ' Response must be between ' + LOFNUM + ' and ' +;
119                 HIFNUM + ', Zero (00) or 99      ' TO ERROR
120             @ 24,8 SAY ERROR
121             DO DELAY
122             SET COLOR TO /W, /W
123             @ 24,0 SAY MESSAGE
124             LOOP
125         ELSE
126             IF MFEAT = '00      ' THEN
127                 EXIT
128             ENDIF
129             IF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
130                 STORE MSITE + MFEAT TO MKEY
131                 USE EQUIP INDEX EQUIPDAT
132                 GO TOP
133                 FIND &MKEY
134                 IF EOF() = .T. THEN
135                     SET COLOR TO W/B, W/B
136                     @ 24,0 SAY SPACE(80)
137                     SET COLOR TO W+/R, W+/R
138                     @ 24,12 SAY ' No record exists for feature number ' +;
139                         MFEAT + ', try again '
140                     DO DELAY
141                     SET COLOR TO /W, /W
142                     @ 24,0 SAY MESSAGE
143                     LOOP
144                 ELSE
145                     EXIT
146                 ENDIF EOF() = .T.
147             ENDIF
148             ENDIF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
149         ENDDO WHILE
150     ENDIF .NOT. (MSITE >= LOSITE .OR. MSITE <= HISITE)

```

Page 4

EQUIPREV.PRG Program Listing

```
151 *
152 SET COLOR TO W/B, W/B
153 @ 24,0 SAY SPACE(80)
154 STORE " At beginning of records for site number " +;
155     MSITE + " " TO TOF
156 STORE " At end of records for site number " + MSITE + " " TO EOF
157 DO WHILE .T.
158     SET COLOR TO R+/B, R+/B
159     @ 6,47 SAY RECNO() PICT "999"
160     STORE FEATURENO TO MFEAT
161     SELECT 2
162     USE DESCRIP INDEX DESCRIP
163     FIND &MFEAT
164     STORE CLIN TO MCLIN
165     STORE DESCRIPT TO MDESCRIP
166     SELECT 1
167     SET COLOR TO /BR, /BR
168     @ 9,20 SAY SITENO PICT "99"
169     @ 9,68 SAY EFFDATE PICT "999999"
170     @ 13,45 SAY MCLIN PICT "9999"
171     @ 14,45 SAY FEATURENO PICT "999999"
172     @ 15,45 SAY MDESCRIP PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
173     @ 16,45 SAY QTY PICT "999"
174     @ 18,50 SAY UNIT PRICE PICT "99999999.99"
175     @ 19,50 SAY MO MAINT PICT "99999999.99"
176     @ 20,53 SAY UNIT INSTA PICT "99999.99"
177     SET COLOR TO R+/B, R+/B
178     STORE "N" TO CHOICE
179     @ 22,68 GET CHOICE PICT "!"
180     READ
181 *
182 * ENSURE THAT THE USER'S PROMPT IS EITHER "N", "P" OR "X"
183 *
184 DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
185     IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
186         SET COLOR TO W+/R, W+/R
187         @ 24,23 SAY " Response must be either N, P or X "
188         DO DELAY
189         STORE "N" TO CHOICE
190     ENDIF
191     SET COLOR TO R+/B, R+/B
192     @ 22,68 GET CHOICE PICT "!"
193     READ
194 ENDDO
195 *
196 * SKIP TO THE NEXT RECORD TO BE REVIEWED
197 *
198 IF CHOICE = "N" THEN
199     IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
200         SKIP
```

```
201      IF EOF() = .T. THEN
202          SKIP - 1
203          SET COLOR TO W+/R, W+/R
204          @ 24,21 SAY EOF
205          DO DELAY
206      ELSE
207          IF .NOT. (SITE NO = MSITE) THEN
208              SKIP - 1
209              SET COLOR TO W+/R, W+/R
210              @ 24,21 SAY EOF
211              DO DELAY
212          ENDIF
213      ENDIF EOF() = .T.
214      ELSE
215          IF RECNO() = LAST_REC THEN
216              GO TOP
217          ELSE
218              SKIP
219          ENDIF
220          ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
221      ENDIF CHOICE = "N"
222      *
223      * SKIP TO THE PREVIOUS RECORD
224      *
225      IF CHOICE = "P" THEN
226          STORE RECNO() TO CURRENTNO
227          IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
228              SKIP - 1
229              IF BOF() = .T. THEN
230                  GOTO CURRENTNO
231                  SET COLOR TO W+/R, W+/R
232                  @ 24,16 SAY TOF
233                  DO DELAY
234              ELSE
235                  IF .NOT. (SITE NO = MSITE) THEN
236                      SKIP
237                      SET COLOR TO W+/R, W+/R
238                      @ 24,16 SAY TOF
239                      DO DELAY
240                  ENDIF
241              ENDIF BOF() = .T.
242          ELSE
243              IF RECNO() = FIRST_REC THEN
244                  GO BOTTOM
245              ELSE
246                  SKIP - 1
247              ENDIF
248              ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
249          ENDIF CHOICE = "P"
250      *
```

```
251 * USER HAS DECIDED TO EXIT THE REVIEW
252 *
253 IF CHOICE = "X"
254     EXIT
255 ENDIF
256 ENDDO WHILE .T.
257 *
258 * RETURN TO CALLING PROGRAM.
259 *
260 RELEASE ALL LIKE M*, CURRENTNO, EOF, ERROR, FIRST_REC, LAST_REC, TOF
261 CLOSE DATABASES
262 RETURN
263 *****
```

```
1 * PROCEDURE EQUIPUPD.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. BUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, SC, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : TO ENABLE THE USER TO MODIFY ANY DATA ELEMENT IN
9 *              : THE EQUIPMENT DATABASE.
10 *
11 * INPUT FILES  : EQUIP.DBF, INDICES: EQUIPPRJ.NDX, EQUIPSIT.NDX
12 *              : EQUIPDAT.NDX, EQUIPSD.NDX
13 *
14 * OUTPUT FILES : EQUIP.DBF, INDICES: EQUIPPRJ.NDX, EQUIPSIT.NDX
15 *              : EQUIPDAT.NDX, EQUIPSD.NDX
16 *
17 * CALLED BY    : EQUIPCMD.PRG
18 *
19 * MODULES CALLED : DELAY.PRG
20 *
21 * GLOBAL VARIABLE: HDATE, HIFNUM, HISITE, LDATE, LOFNUM, LOSITE
22 *
23 * LOCAL VARIABLES: MEFFDATE, MSITE, MSITE, MFEAT, MPRICE,
24 *                 MMAINT, MINSTALL, MQTY, MESSAGE
25 *                 ACCEPT, CHOICE, CURRENINO, EOF, ERROR, FIRST_REC,
26 *                 INTRO, LAST_REC, TOF
27 *
28 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
29 *
30 * CASE SELECTION = 1      UPDATE EXISTING RECORDS
31 *
32 * USE EQUIPMENT DATABASE USING THE SITE NUMBER INDEX, BUT UPDATING
33 * ALL EQUIP FILE RELATED INDICES, ASK THE USER TO INPUT A SITE
34 * NUMBER THEN START UPDATING FROM THAT POINT.
35 *
36 SET ESCAPE OFF
37 SET TALK OFF
38 USE EQUIP
39 GO TOP
40 SET COLOR TO W+/B, W+/B, B
41 CLEAR
42 IF EOF() = .T. THEN
43     SET COLOR TO W+/R, W+/R
44     @ 13,24 SAY " The EQUIPMENT Database is EMPTY! "
45     DO DELAY
46     RETURN
47 ENDIF
48 ?? FLASH + "S.EQUIPUPD.SCR/"
49 @ 24,0 SAY SPACE(80)
50 STORE "Enter 00 to start at TOF, 99 to start at EOF, or a site " +;
```

```

51      "number between " + LOSITE + " and " + HISITE + " " TO MESSAGE
52 SET COLOR TO /W, /W
53 @ 24,0 SAY MESSAGE
54 STORE '88' TO MSITE
55 DO WHILE .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
56     SET COLOR TO /BR, /BR
57     STORE '00' TO MSITE
58     @ 8,20 GET MSITE PICT '99'
59     READ
60     IF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
61         SET COLOR TO W/B, W/B
62         @ 24,0 SAY SPACE(80)
63         SET COLOR TO W+/R, W+/R
64         STORE ' Response must be between ' + LOSITE + ' and ' + HISITE + ;
65             ', Zero (00) or 99 ' TO ERROR
66         @ 24,13 SAY ERROR
67         DO DELAY
68         SET COLOR TO /W, /W
69         @ 24,0 SAY MESSAGE
70         LOOP
71     ELSE
72         IF (MSITE = '00' .OR. MSITE = '99') THEN
73             USE EQUIP
74             IF MSITE = '00' THEN
75                 GO BOTTOM
76                 STORE RECNO() TO LAST_REC
77                 GO TOP
78                 STORE RECNO() TO FIRST_REC
79             ELSE
80                 IF MSITE = '99' THEN
81                     GO TOP
82                     STORE RECNO() TO FIRST_REC
83                     GO BOTTOM
84                     STORE RECNO() TO LAST_REC
85                 ENDIF MSITE = '99'
86             ENDIF MSITE = '00'
87         ELSE
88             USE EQUIP INDEX EQUIPSIT, EQUIPPRJ, EQUIPDAT, EQUIPSD
89             GO TOP
90             FIND &MSITE
91             IF EOF() = .T. THEN
92                 SET COLOR TO W/B, W/B
93                 @ 24,0 SAY SPACE(80)
94                 STORE " No records exist for site number " + MSITE + ;
95                     ", try again " TO ERROR
96                 SET COLOR TO W+/R, W+/R
97                 @ 24,16 SAY ERROR
98                 DO DELAY
99                 SET COLOR TO /W, /W
100                @ 24,0 SAY MESSAGE

```



```

101         STORE '88' TO MSITE
102     ENDIF
103     ENDIF
104     ENDIF
105     ENDDO WHILE .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
106     *
107     STORE SPACE(10) + 'Enter "00  " to start at TOF or a six digit ' +;
108     'feature number' + SPACE(10) TO MESSAGE
109     IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
110         SET COLOR TO /W, /W
111         @ 24,0 SAY MESSAGE
112         DO WHILE .T.
113             SET COLOR TO /BR, /BR
114             STORE '00  ' TO MFEAT
115             @ 11,45 GET MFEAT PICT '999999'
116             READ
117             IF .NOT. ((MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) .OR.;
118                 MFEAT = '00  ' .OR. MFEAT = '99  ')
119                 SET COLOR TO W/B, W/B
120                 @ 24,0 SAY SPACE(80)
121                 SET COLOR TO W+/R, W+/R
122                 STORE ' Response must be between ' + LOFNUM + ' and ' +;
123                 HIFNUM + ', Zero (00) or 99  ' TO ERROR
124                 @ 24,8 SAY ERROR
125                 DO DELAY
126                 SET COLOR TO /W, /W
127                 @ 24,0 SAY MESSAGE
128                 LOOP
129             ELSE
130                 IF MFEAT = '00  ' THEN
131                     EXIT
132                 ENDIF
133                 IF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
134                     STORE MSITE + MFEAT TO MKEY
135                     USE EQUIP INDEX EQUIPDAT
136                     GO TOP
137                     FIND &MKEY
138                     IF EOF() = .T. THEN
139                         SET COLOR TO W/B, W/B
140                         @ 24,0 SAY SPACE(80)
141                         SET COLOR TO W+/R, W+/R
142                         STORE ' No record exists for feature number ' +;
143                         MFEAT + ', try again ' TO ERROR
144                         @ 24,12 SAY ERROR
145                         DO DELAY
146                         SET COLOR TO /W, /W
147                         @ 24,0 SAY MESSAGE
148                         LOOP
149                     ELSE
150                         EXIT

```

Page 4

EQUIPUPD.PRG Program Listing

```

151         ENDIF EOF() = .T.
152         ENDIF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
153     ENDIF
154     ENDDO WHILE .T.
155 ENDIF .NOT. (MSITE >= LOSITE .OR. MSITE <= HISITE)
156 *
157 SET COLOR TO W/B, W/B
158 @ 24,0 SAY SPACE(80)
159 STORE " At beginning of records for site number " +;
160     MSITE + " " TO TOF
161 STORE " At end of records for site number " + MSITE + " " TO EOF
162 STORE SPACE(16) + 'Press "Page Down" key to terminate record update' +;
163     SPACE(16) TO MESSAGE
164 STORE 1 TO INTRO
165 DO WHILE .T.
166     SET COLOR TO /W, /W
167     @ 24,0 SAY MESSAGE
168 *
169 * STORING THE OLD RECORD TO A WORK RECORD AREA. THE M PREFIX
170 * INDICATES MEMORY VARIABLES DISTINGUISHING THEM FROM THEIR
171 * CORRESPONDING DATABASE FIELDS.
172 *
173 STORE UNIT PRICE TO MPRICE
174 STORE MO_MAINT TO MMAINT
175 STORE UNIT_INSTA TO MINSTALL
176 STORE QTY TO MQTY
177 STORE FEATURENO TO MFEAT
178 SELECT 2
179 USE DESCRIP INDEX DESCRIP
180 FIND &MFEAT
181 STORE DESCPT TO MDESCPT
182 SELECT 1
183 *
184 * INFORM THE USER OF HOW TO TERMINATE THE UPDATE OF A RECORD
185 *
186 IF INTRO = 1 THEN
187     STORE 0 TO INTRO
188     ?? FLASH + "W.EQUIPUPD/"
189     SET CONSOLE OFF
190     WAIT TO ANS
191     SET CONSOLE ON
192 ENDIF
193 *
194 SET COLOR TO R+/B, R+/B
195 @ 5,47 SAY RECNO() PICT "999"
196 SET COLOR TO /BR, /BR
197 @ 8,20 SAY SITENO PICT "99"
198 @ 8,68 SAY EFFDATE PICT "999999"
199 @ 11,45 SAY MFEAT PICT "999999"
200 @ 12,45 SAY MDESCPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

```

```
201 @ 13,45 SAY MQTY PICT "999"
202 @ 15,50 GET MPRICE PICT "99999999.99"
203 @ 16,50 GET MMAINT PICT "99999999.99"
204 @ 17,53 GET MINSTALL PICT "99999.99"
205 READ
206 SET COLOR TO W/B, W/B
207 @ 24,0 SAY SPACE(80)
208 *
209 IF .NOT. (QTY=MQTY .AND. UNIT_PRICE=MPRICE .AND.;
210          MO_MAINT=MMAINT .AND. UNIT_INSTA=MINSTALL) THEN
211 *
212 *   ASK THE USER IF HE/SHE DESIRES TO ACCEPT THE CHANGES.
213 *
214 SET COLOR TO W+/B, W+/B
215 @ 19,12 SAY "Do you want to accept the changes? (Yes or No): "
216 SET COLOR TO R+/B, R+/B
217 @ 19,49 SAY "Y"
218 @ 19,56 SAY "N"
219 STORE "N" TO ACCEPT
220 @ 19,62 GET ACCEPT PICT "!"
221 READ
222 *
223 *   ENSURE THAT THE USER'S PROMPT IS EITHER "Y" OR "N"
224 *
225 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
226     IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
227         SET COLOR TO W/B, W/B
228         @ 24,0 SAY SPACE(80)
229         SET COLOR TO W+/R, W+/R
230         @ 24,24 SAY " Response must be either N or Y "
231         DO DELAY
232         STORE "N" TO ACCEPT
233     ENDIF
234     SET COLOR TO R+/B, R+/B
235     @ 19,62 GET ACCEPT PICT "!"
236     READ
237 ENDDO
238 @ 19,62 SAY " "
239 *
240 IF ACCEPT = "Y" THEN
241     REPLACE UNIT_PRICE WITH MPRICE
242     REPLACE MO_MAINT WITH MMAINT
243     REPLACE UNIT_INSTA WITH MINSTALL
244     REPLACE QTY WITH MQTY
245 ENDIF
246 ENDIF
247 *
248 SET COLOR TO W/B, W/B
249 @ 19,10 SAY SPACE(60)
250 SET COLOR TO R+/B, R+/B
```

```
251 STORE "N" TO CHOICE
252 @ 21,68 GET CHOICE PICT "!"
253 READ
254 *
255 * ENSURE THAT THE USER'S PROMPT IS EITHER "N", "P" OR "X"
256 *
257 DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
258     IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
259         SET COLOR TO W/B, W/B
260         @ 24,0 SAY SPACE(80)
261         SET COLOR TO W+/R, W+/R
262         @ 24,23 SAY " Response must be either N, P or X "
263         DO DELAY
264         STORE "N" TO CHOICE
265     ENDIF
266     SET COLOR TO R+/B, R+/B
267     @ 21,68 GET CHOICE PICT "!"
268     READ
269 ENDDO
270 *
271 * SKIP TO THE NEXT RECORD TO BE REVIEWED
272 *
273 IF CHOICE = "N" THEN
274     IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
275         SKIP
276         IF EOF() = .T. THEN
277             SKIP - 1
278             SET COLOR TO W+/R, W+/R
279             @ 24,21 SAY EOF
280             DO DELAY
281         ELSE
282             IF .NOT. (SITE NO = MSITE) THEN
283                 SKIP - 1
284                 SET COLOR TO W+/R, W+/R
285                 @ 24,21 SAY EOF
286                 DO DELAY
287             ENDIF
288             ENDIF EOF() = .T.
289         ELSE
290             IF RECNO() = LAST_REC THEN
291                 GO TOP
292             ELSE
293                 SKIP
294             ENDIF
295         ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
296     ENDIF CHOICE = "N"
297 *
298 * SKIP TO THE PREVIOUS RECORD
299 *
300 IF CHOICE = "P" THEN
```

Page 7

EQUIPUPD.PRG Program Listing

```

301 STORE RECNO() TO CURRENTNO
302 IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
303     SKIP - 1
304     IF BOF() = .T. THEN
305         GOTO CURRENTNO
306         SET COLOR TO W+/R, W+/R
307         @ 24,16 SAY TOF
308         DO DELAY
309     ELSE
310         IF .NOT. (SITENO = MSITE) THEN
311             SKIP
312             SET COLOR TO W+/R, W+/R
313             @ 24,16 SAY TOF
314             DO DELAY
315         ENDIF
316     ENDIF BOF() = .T.
317 ELSE
318     IF RECNO() = FIRST_REC THEN
319         GO BOTTOM
320     ELSE
321         SKIP - 1
322     ENDIF
323     ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE'58')
324     ENDIF CHOICE = "P"
325 *
326 * USER HAS DECIDED TO EXIT THE REVIEW
327 *
328 IF CHOICE = "X"
329     EXIT
330 ENDIF
331 *
332 ENDDO WHILE .T.
333 *
334 * RETURN TO CALLING PROGRAM.
335 *
336 RELEASE ALL LIKE M*, ACCEPT, CHOICE, CURRENTNO, EOF, ERROR,;
337     FIRST_REC, INTRO, LAST_REC, TOF
338 CLOSE DATABASES
339 RETURN
340 *****

```

Page 1

MAINMENU.PRG Program Listing

```

1  * PROCEDURE MAINMENU.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              LCDR WINSTON H. BUCKLEY, SC, USN
5  *              LCDR ROBERT F. BRADO, USN
6  *              LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER THE CHOICE OF LOADING A NEW DELIVERY,
9  *              ORDER, MAINTAINING THE EQUIPMENT, MANUAL, AND
10 *              SERIAL NUMBER DATA BASES OR GETTING A SERIES OF
11 *              REPORTS FROM THESE UPDATED DATABASES.
12 *
13 * INPUT FILES   : NONE.
14 *
15 * OUTPUT FILES  : NONE.
16 *
17 * CALLED BY     : SELECTOR.PRG
18 *
19 * MODULES CALLED : NEWDOCMD.PRG, EQUIPCMD.PRG, MANULCMD.PRG,
20 *              SERNOCMD.PRG, REPORCMD.PRG, DESPMOD.PRG,
21 *              CONFMOD.PRG, DELAY.PRG, MAINTDO.PRG, MKLABELS.PRG
22 *
23 * GLOBAL VARIABLES : HIDATE, HIFNUM, HISITE, LODATE, LOFNUM, LOSITE
24 *
25 * LOCAL VARIABLES : ANS
26 *
27 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
28 *
29 * DBASE PROGRAM CONFIGURATION VARIABLES:
30 *
31 SET BELL OFF
32 SET CONSOLE ON
33 SET INTENSITY OFF
34 SET SCOREBOARD OFF
35 SET TALK OFF
36 PUBLIC HIDATE, HIFNUM, HISITE, LODATE, LOFNUM, LOSITE
37 *
38 * INITIALIZE THE PUBLIC VARIABLES
39 *
40 STORE '991231' TO HIDATE
41 STORE '994001' TO HIFNUM
42 STORE '58' TO HISITE
43 STORE '840101' TO LODATE
44 STORE '000101' TO LOFNUM
45 STORE '01' TO LOSITE
46 *
47 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE USER'S CHOICE.
48 *
49 STORE '1' TO ANS
50 DO WHILE .T.

```

Page 2

MAINMENU.PRG Program Listing

```
51 FLASH = CHR(145)
52 SET COLOR TO W/B, W/B, B
53 ?? FLASH + "S.MAINMENU.SCR/"
54 @ 24,0 SAY SPACE (80)
55 SET COLOR TO R+/B, R+/B
56 @ 22,53 GET ANS PICT "9"
57 READ
58 *
59 * PERFORM APPROPRIATE TASK BASED ON THE USER'S CHOICE.
60 *
61 DO CASE
62 *
63 * CALL THE NEW DELIVERY ORDER LOAD COMMAND PROGRAM.
64 CASE ANS = "1"
65 DO NEWDOCMD
66 STORE "1" TO ANS
67 *
68 * CALL THE EQUIPMENT FILE MAINTENANCE COMMAND PROGRAM.
69 CASE ANS = "2"
70 DO EQUIPCMD
71 STORE "2" TO ANS
72 *
73 * CALL THE DESCRIPTION FILE MAINTENANCE COMMAND PROGRAM.
74 CASE ANS = "3"
75 DO DESPMOD
76 STORE "3" TO ANS
77 *
78 * CALL THE SITE CONFIGURATION FILE MAINTENANCE COMMAND PROGRAM.
79 CASE ANS = "4"
80 DO CONFMOD
81 STORE "4" TO ANS
82 *
83 * CALL THE MANUAL FILE MAINTENANCE COMMAND PROGRAM.
84 CASE ANS = "5"
85 DO MANULCMD
86 STORE "5" TO ANS
87 *
88 * CALL THE SERIAL NUMBER MAINTENANCE COMMAND PROGRAM.
89 CASE ANS = "6"
90 DO SERNOCMD
91 STORE "6" TO ANS
92 *
93 * CALL THE REPORTS GENERATION COMMAND PROGRAM.
94 CASE ANS = "7"
95 DO REPORCMD
96 STORE "7" TO ANS
97 *
98 * CALL THE MAINTENANCE DELIVERY ORDER GENERATION PROGRAM
99 CASE ANS = "8"
100 DO MAINTDO
```

Page 3

MAINMENU.PRG Program Listing

```

101      STORE "8" TO ANS
102      *
103      *      CALL THE MAILING LABELS GENERATION PROGRAM
104      CASE ANS = "9"
105          DO MKLABELS
106          STORE "9" TO ANS
107      *
108      *      RETURN THE USER TO SELECTOR PROGRAM CONTROL.
109      CASE ANS = "0"
110          CLOSE DATABASES
111          RETURN
112      *
113      ENDCASE
114      *
115      *      CONTINUE PROCESSING LOOP CONTROL CHECK.
116      *
117      ENDDO WHILE .T.
118      *****

```


Page 1

MAINTDO.PRG Program Listing

```

1 * PROCEDURE MAINTDO.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. EUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : PROVIDE THE INPUTS FOR A MAINTENANCE DELIVERY
9 *              : ORDER, WHICH WILL BW IMPORTED INTO LOTUS 1-2-3.
10 *
11 * INPUT FILES  : EQUIP.DBF, DESCRIP.DBF, DECSRIP.NDX, TEMPONE.DBF
12 *              : EFEBT.NDX, TEMOTWO.DBF, TEMPTIRE.DBF, TEMPFOUR.DBF
13 *
14 * OUTPUT FILE  : NEWDO.PRN
15 *
16 * CALLED BY    : MAINMENU.PRG
17 *
18 * MODULES CALLED : DELAY.PRG
19 *
20 * GLOBAL VARIABLE: HISITE, LOSITE
21 *
22 * LOCAL VARIABLES: ERROR, MESSAGE, MSITE, NOFIND, RATES, SITES
23 *
24 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
25 *
26 SET ESCAPE OFF
27 SET TALK OFF
28 SET COLOR TO W+/B, W+/B, B
29 CLEAR
30 ?? FLASH + "S.MAINTDO.SCR/"
31 @ 24,0 SAY SPACE(80)
32 STORE " Enter the number of the site for which the maintenance is " +;
33 "to be performed " TO SITES
34 STORE SPACE(20) + " Enter the Discount and Escalation Rates " +;
35 SPACE(20) TO RATES
36 SET COLOR TO /BR, /BR
37 @ 20,57 SAY " NEWDO.PRN "
38 *
39 * OBTAIN THE NUMBER OF THE SITE TO RECEIVE THE MAINTENANCE FROM THE USER
40 *
41 USE EQUIP INDEX EQUIPSIT.NDX
42 *
43 DO WHILE .T.
44 SET COLOR TO /W, /W
45 @ 24,0 SAY SITES
46 SET COLOR TO R+/B, R+/B
47 STORE LOSITE TO MSITE
48 @ 04,65 GET MSITE PICT "99"
49 READ
50 IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN

```

```

51      SET COLOR TO W/B, W/B
52      @ 24,0 SAY SPACE(80)
53      SET COLOR TO W+/R, W+/R
54      STORE ' Response must be between ' + LOSITE +;
55          ' and ' + HISITE + ' ' TO ERROR
56      @ 24,22 SAY ERROR
57      DO DELAY
58      LOOP
59  ELSE
60      GO TOP
61      FIND &MSITE
62      IF EOF() = .T. THEN
63          SET COLOR TO W/B, W/B
64          @ 24,0 SAY SPACE(80)
65          SET COLOR TO W+/R, W+/R
66          STORE " No records for site number " + MSITE +;
67              " exist, try again " TO MESSAGE
68          @ 24,16 SAY MESSAGE
69          DO DELAY
70          LOOP
71      ELSE
72          EXIT
73      ENDIF EOF() = .T.
74      ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
75  ENDDO WHILE .T.
76  *
77  * ENSURE THAT TEMPORARY DATABASES DO NOT EXIST, IF SO ERASE THEM
78  *
79  SET CONSOLE OFF
80  ERASE TEMPONE.DBF
81  ERASE TEMPONE.NDX
82  ERASE TEMPTWO.DBF
83  ERASE TEMPTHRE.DBF
84  ERASE TEMPFOUR.DBF
85  SET CONSOLE ON
86  *
87  * INFORM THE USER THAT THERE WILL BE A SLIGHT DELAY
88  *
89  SET COLOR TO W+/R, W+/R
90  STORE SPACE(10) + "Creating a temporary database and index. " +;
91      "PLEASE BE PATIENT " + SPACE(10) TO MESSAGE
92  @ 24,0 SAY MESSAGE
93  COPY TO TEMPONE.DBF WHILE SITENO = "&MSITE"
94  USE TEMPONE
95  INDEX ON FEATURENO TO TEMPONE
96  TOTAL ON FEATURENO TO TEMPTWO.DBF FIELDS QTY WHILE FEATURENO <> 'XXXXXX'
97  *
98  * OBTAIN THE DISCOUNT AND ESCALATION RATES FROM THE USER
99  *
100 SET COLOR TO /W, /W

```

Page 3

MAINTDO.PRG Program Listing

```
101 @ 24,0 SAY RATES
102 STORE "0.000" TO LCNHWRATE
103 STORE "0.000" TO LCNSWRATE
104 STORE "0.000" TO SNETSWRATE
105 STORE "0.000" TO UPLIFT
106 SET COLOR TO /BR, /BR
107 @ 14,61 GET LCNHWRATE PICT "9.999"
108 @ 15,61 GET LCNSWRATE PICT "9.999"
109 @ 16,61 GET SNETSWRATE PICT "9.999"
110 @ 17,61 GET UPLIFT PICT "9.999"
111 READ
112 *
113 * ASK TO USER TO VERIFY THAT HE/SHE WANTS TO CONTINUE
114 *
115 SET COLOR TO W+/B, W+,B
116 @ 24,0 SAY SPACE(80)
117 @ 22,22 SAY "Do you want to Continue or eXit? "
118 SET COLOR TO R+/B, R+/B
119 @ 22,37 SAY "C"
120 @ 22,50 SAY "X"
121 STORE "C" TO CHOICE
122 @ 22,56 GET CHOICE PICT "!"
123 READ
124 *
125 * ENSURE THAT THE USER'S RESPONSE IS EITHER "N", "P" OR "X"
126 *
127 DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
128 IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
129 SET COLOR TO W+/R, W+/R
130 @ 24,24 SAY " Response must be either C or X "
131 DO DELAY
132 STORE "C" TO CHOICE
133 ENDIF .NOT. (CHOICE = "C" .OR. CHOICE = "X")
134 SET COLOR TO R+/B, R+/B
135 @ 22,56 GET CHOICE PICT "!"
136 READ
137 ENDDO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
138 *
139 SET COLOR TO W/B, W/B
140 @ 22,20 SAY SPACE(50)
141 IF CHOICE = "C" THEN
142 STORE 1 + VAL(LCNHWRATE) TO LCNHWRATE
143 STORE 1 + VAL(LCNSWRATE) TO LCNSWRATE
144 STORE 1 + VAL(SNETSWRATE) TO SNETSWRATE
145 STORE 1 + VAL(UPLIFT) TO UPLIFT
146 ELSE
147 SET CONSOLE OFF
148 CLOSE DATABASES
149 ERASE TEMPONE.DBF
150 ERASE TEMPONE.NDX
```

Page 4

MAINTDO.PRG Program Listing

```

151 ERASE TEMPTWO.DBF
152 ERASE TEMPTHRE.DBF
153 ERASE TEMPFOUR.DBF
154 SET CONSOLE ON
155 SET COLOR TO W/B, W/B
156 @ 24,0 SAY SPACE(80)
157 RELEASE ERROR, MESSAGE, MSITE, NOFIND, RATES, SITES
158 RETURN
159 ENDIF
160 *
161 * INFORM THE USER THAT THERE WILL BE A SLIGHT DELAY
162 *
163 SET COLOR TO W+/R, W+/R
164 STORE " Creating the MAINTENANCE DELIVERY ORDER may take up to 10 " +;
165 "minutes. PLEASE WAIT " TO MESSAGE
166 @ 24,0 SAY MESSAGE
167 SELECT 1
168 USE TEMPTWO
169 SELECT 2
170 USE DESCRIP
171 SELECT TEMPTWO
172 JOIN WITH DESCRIP TO TEMPTHRE FOR FEATURENO = DESCRIP->FEATURENO
173 SELECT 3
174 USE TEMPTHRE
175 GO TOP
176 REPLACE ALL MO_MAINT WITH BASEMAINT*LCNSWRATE FOR FEATURENO > "320100" .AND.;
177 FEATURENO < "420400"
178 GO TOP
179 REPLACE ALL MO_MAINT WITH BASEMAINT*LCNSWRATE FOR FEATURENO = "550801"
180 REPLACE ALL MO_MAINT WITH BASEMAINT*LCNSWRATE FOR FEATURENO = "550901"
181 REPLACE ALL MO_MAINT WITH BASEMAINT*LCNSWRATE FOR FEATURENO = "551001"
182 REPLACE ALL MO_MAINT WITH BASEMAINT*LCNSWRATE FOR FEATURENO = "551101"
183 REPLACE ALL MO_MAINT WITH BASEMAINT*LCNSWRATE FOR FEATURENO = "551201"
184 REPLACE ALL MO_MAINT WITH BASEMAINT*LCNSWRATE FOR FEATURENO = "551301"
185 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "550710"
186 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "550711"
187 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "550803"
188 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "550903"
189 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551003"
190 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551103"
191 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551203"
192 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551303"
193 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551304"
194 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551403"
195 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551500"
196 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551501"
197 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551502"
198 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551503"
199 REPLACE ALL MO_MAINT WITH BASEMAINT*SNETSWRATE FOR FEATURENO = "551504"
200 SELECT 4

```

Page 5

MAINTDO.PRG Program Listing

```
201 USE TED
202 COPY STRUCTURE TO TEMPF0UR
203 CLOSE DATABASES
204 USE TEMPF0UR
205 APPEND FROM TEMPTHRE
206 GO TOP
207 REPLACE ALL MAINT_MOS WITH 12
208 REPLACE ALL MAINT_FAC WITH UPLIFT
209 REPLACE ALL MAINT_FAC WITH 1 FOR FEATURENO = "550801"
210 REPLACE ALL MAINT_FAC WITH 1 FOR FEATURENO = "550901"
211 REPLACE ALL MAINT_FAC WITH 1 FOR FEATURENO = "551001"
212 REPLACE ALL MAINT_FAC WITH 1 FOR FEATURENO = "551101"
213 REPLACE ALL MAINT_FAC WITH 1 FOR FEATURENO = "551201"
214 REPLACE ALL MAINT_FAC WITH 1 FOR FEATURENO = "551301"
215 REPLACE ALL TOT_MAINT WITH MAINT_FAC*MO_MAINT*MAINT_MOS
216 REPLACE ALL COMP_DT_CR WITH (((UNIT_PRICE + UNIT_INSTA)/48) +;
217     (MO_MAINT * MAINT_FAC)) * .005
218 REPLACE ALL SYS_DT_CR WITH (QTY*MO_MAINT*MAINT_FAC)
219 REPLACE ALL TOT_MAINT WITH TOT_MAINT*QTY FOR FEATURENO > "010200" .AND.;
220     FEATURENO < "510101"
221 REPLACE ALL UNIT_PRICE WITH 0
222 REPLACE ALL TOT_PRICE WITH 0
223 REPLACE ALL UNIT_INSTA WITH 0
224 REPLACE ALL TOT_INSTAL WITH 0
225 COPY TO NEWDO.PRN DELIMITED
226 *
227 *   ERASE ALL TEMPORARY DATABASES AND INDICES CREATED DURING THE PROGRAM
228 *
229 SET CONSOLE OFF
230 CLOSE DATABASES
231 ERASE TEMPONE.DBF
232 ERASE TEMPTWO.DBF
233 ERASE TEMPTHRE.DBF
234 ERASE TEMPF0UR.DBF
235 ERASE TEMPONE.NDX
236 SET CONSOLE ON
237 *
238 *   RETURN TO CALLING PROGRAM
239 *
240 SET COLOR TO W/B, W/B
241 @ 24,0 SAY SPACE(80)
242 RELEASE ERROR, MESSAGE, MSITE, NOFIND, RATES, SITES
243 RETURN
244 *****
```

Page 1

MANULADD.PRG Program Listing

```
1 * PROCEDURE MANULADD.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. BUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : ADD NEW MANUALS TO THE MANUAL DATABASE FILE.
9 *
10 * INPUT FILES  : MANUAL.DBF, MANULSIT.NDX
11 *
12 * CALLED BY    : MANULCMD.PRG
13 *
14 * MODULES CALLED : DELAY.PRG
15 *
16 * GLOBAL VARIABLE: HIFNUM, HISITE, LOFNUM, LOSITE
17 *
18 * LOCAL VARIABLES: ACCEPT, CHOICE, ERROR, FEATURES, GETOUT,
19 *                 MCLIN, MANDESCRPT, MDESCRPT, MESSAGE, MFEAT,
20 *                 MSITE, NOFIND, NOSITE, SITES
21 *
22 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
23 *
24 * CASE SELECTION = 1      ADD A NEW MANUAL DESCRIPTION
25 *
26 SET ESCAPE OFF
27 SET TALK OFF
28 USE MANUAL
29 GO TOP
30 SET COLOR TO W+/B, W+/B, B
31 CLEAR
32 IF EOF() = .T. THEN
33     SET COLOR TO W+/R, W+/R
34     @ 13,25 SAY " The MANUALS Database is EMPTY! "
35     DO DELAY
36     RETURN
37 ENDIF
38 SELECT 1
39 USE MANUAL INDEX MANULSIT
40 *
41 ?? FLASH + "S.MANUALS.SCR/"
42 @ 24,0 SAY SPACE(80)
43 @ 22,10 SAY SPACE(60)
44 SET COLOR TO GR+/B, GR+/B
45 @ 6,28 SAY "      Last "
46 SET COLOR TO R+/ , R+/
47 @ 3,26 SAY " MANUAL ADDITION FORMAT "
48 SET COLOR TO W+/B, W+/B
49 @ 22,23 SAY "Enter C to continue or X to exit: "
50 SET COLOR TO R+/B, R+/B
```

```
51 @ 22,29 SAY "C"
52 @ 22,46 SAY "X"
53 *
54 * GENERATE STATUS MESSAGES
55 *
56 STORE ' Enter a Site Number between ' + LOSITE + ' and ' +;
57 HISITE + ' for the Manual Description Addition ' TO SITES
58 STORE ' Enter a Feature Number ( ' + LOFNUM + ' - ' + HIFNUM + ' ) ' +;
59 'for the Manual Description Addition ' TO FEATURES
60 STORE SPACE(20) + 'Enter the Manual Description to be Added' +;
61 SPACE(20) TO MANDESCRPT
62 *
63 DO WHILE .T.
64 SET COLOR TO R+/B, R+/B
65 @ 6,47 SAY RECNO() PICT "9999"
66 *
67 * CLEAR SCREEN AND SET INITIAL VALUES FOR VARIABLES TO BE
68 * ADDED TO THE FILE. THE M PREFIX INDICATES MEMORY VARIABLES
69 * DISTINGUISHING THEM FROM THEIR CORRESPONDING DATABASE FIELDS.
70 *
71 STORE ' 'TO MFEAT
72 STORE ' ' TO MMANDESC
73 *
74 SET COLOR TO /W, /W
75 @ 24,0 SAY SITES
76 *
77 * ENSURE THAT THE SITE NUMBER IS A VALID SITE
78 *
79 DO WHILE .T.
80 SET COLOR TO /BR, /BR
81 STORE LOSITE TO MSITE
82 @ 9,45 GET MSITE PICT '99'
83 READ
84 IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
85 SET COLOR TO W/B, W/B
86 @ 24,0 SAY SPACE(80)
87 SET COLOR TO W+/R, W+/R
88 STORE ' Response must be between ' + LOSITE + ' and ' +;
89 HISITE + ' ' TO ERROR
90 @ 24,22 SAY ERROR
91 DO DELAY
92 SET COLOR TO /W, /W
93 @ 24,0 SAY SITES
94 LOOP
95 ELSE
96 GO TOP
97 FIND &MSITE
98 IF EOF() = .T. THEN
99 SET COLOR TO W/B, W/B
100 @ 24,0 SAY SPACE(80)
```

Page 3

MANULADD.PRG Program Listing

```

101      SET COLOR TO W+/R, W+/R
102      STORE " No records exist for site " + MSITE +;
103      ", try another site " to NOSITE
104      @ 24,16 SAY NOSITE
105      DO DELAY
106      SET COLOR TO /W, /W
107      @ 24,0 SAY SITES
108      STORE "99" TO MSITE
109      LOOP
110      ELSE
111      EXIT
112      ENDIF EOF() = .T.
113      ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
114      ENDDO WHILE .T.
115      *
116      GO BOTTOM
117      SET COLOR TO /W, /W
118      @ 24,0 SAY FEATURES
119      SET COLOR TO /BR, /BR
120      STORE 0 TO NOFIND
121      STORE "N" TO GETOUT
122      *
123      * ENSURE THAT THE FEATURE IS A VALID FEATURE
124      *
125      DO WHILE .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
126      *
127      * IF THE USER HAS MADE THREE ATTEMPTS TO SPECIFY A VALID .PRN FILE
128      * NAME AND HAS NOT BEEN SUCCESSFUL, ASK HIM/HER IF THEY DESIRE
129      * EXIT THIS PROCESS.
130      *
131      IF NOFIND = 3 THEN
132      SET COLOR TO W+/B, W+/B
133      @ 19,15 SAY " Do you want to exit this process? (Yes or No): "
134      SET COLOR TO R+/B, R+/B
135      @ 19,51 SAY "Y"
136      @ 19,58 SAY "N"
137      STORE "Y" TO GETOUT
138      @ 19,63 GET GETOUT PICT "!"
139      READ
140      *
141      DO WHILE .NOT. (GETOUT = "N" .OR. GETOUT = "Y")
142      IF .NOT. (GETOUT = "N" .OR. GETOUT = "Y") THEN
143      SET COLOR TO W+/R, W+/R
144      @ 24,24 SAY " Response must be either N or Y "
145      DO DELAY
146      STORE "Y" TO GETOUT
147      ENDIF
148      SET COLOR TO R+/B, R+/B
149      @ 19,63 GET GETOUT PICT "!"
150      READ

```



```
151      ENDDO
152  *
153      SET COLOR TO W/B, W/B
154      @ 19,10 SAY SPACE(65)
155      IF GETOUT = "Y" THEN
156          EXIT
157      ELSE
158          STORE 0 TO NOFIND
159          SET COLOR TO /W, / W
160          @ 24,0 SAY FEATURES
161          LOOP
162      ENDIF
163  ENDIF
164      IF GETOUT = "Y" THEN
165          EXIT
166      ENDIF
167      SET COLOR TO /BR, /BR
168      STORE LOFNUM TO MFEAT
169      @ 12,45 GET MFEAT PICT '999999'
170      READ
171  *
172  *      ENSURE THAT THE FEATURE NUMBER ENTERED BY THE USER IS VALID
173  *
174      IF .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) THEN
175          SET COLOR TO W/B, W/B
176          @ 24,0 SAY SPACE(80)
177          SET COLOR TO W+/R, W+/R
178          STORE ' Response must be between ' + LOFNUM + ;
179              ' and ' + HIFNUM + ' ' TO ERROR
180          @ 24,18 SAY ERROR
181          DO DELAY
182          SET COLOR TO /W, /W
183          @ 24,0 SAY FEATURES
184      ELSE
185          SELECT 2
186          USE EQUIP INDEX EFEAT
187          GO TOP
188          FIND &MFEAT
189          IF EOF() = .T. THEN
190              NOFIND = NOFIND + 1
191              SET COLOR TO W/B, W/B
192              @ 24,0 SAY SPACE(80)
193              SET COLOR TO W+/R, W+/R
194              STORE " Feature Number " + MFEAT + ;
195                  " does not exist, try again " TO MESSAGE
196              IF NOFIND < 3 THEN
197                  @ 24,16 SAY MESSAGE
198                  DO DELAY
199                  SET COLOR TO /W, /W
200                  @ 24,0 SAY FEATURES
```

Page 5

MANULADD.PRG Program Listing

```
201         ENDIF
202         STORE "999999" TO MFEAT
203         SELECT 1
204         ENDIF EOF() = .T.
205         ENDIF .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
206         ENDDO WHILE .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
207     *
208     SET COLOR TO W+/B, W+/B
209     @ 24,0 SAY SPACE(80)
210     IF GETOUT = "Y" THEN
211         EXIT
212     ENDIF
213     SELECT 3
214     USE DESCRIP INDEX DESCRIP
215     GO TOP
216     FIND &MFEAT
217     STORE CLIN TO MCLIN
218     STORE DESCRIPT TO MDESCIPT
219     SELECT 1
220     SET COLOR TO /BR, /BR
221     @ 13,45 SAY MCLIN PICT "9999"
222     @ 14,45 SAY MDESCIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
223 *
224     SET COLOR TO /W, /W
225     @ 24,0 SAY MANDESCRPT
226     SET COLOR TO /BR, /BR
227     @ 17,45 GET MMANDESC PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
228     READ
229     SET COLOR TO W/B, W/B
230     @ 24,0 SAY SPACE(80)
231 *
232     IF .NOT. (MANLDESC = MMANDESC) THEN
233 *
234 *         ASK THE USER IF HE/SHE DESIRES TO ACCEPT THE CHANGES
235 *
236         SET COLOR TO W+/B, W+/B
237         @ 20,12 SAY "Do you want to accept the change? (Yes or No):"
238         SET COLOR TO R+/B, R+/B
239         @ 20,49 SAY "Y"
240         @ 20,56 SAY "N"
241         STORE "N" TO ACCEPT
242         @ 20,62 GET ACCEPT PICT "!"
243         READ
244 *
245 *         ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
246 *
247         DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
248             IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
249                 SET COLOR TO W+/R, W+/R
250                 @ 24,24 SAY " Response must be either N or Y "
```

Page 6

MANULADD.PRG Program Listing

```

251         DO DELAY
252         STORE "N" TO ACCEPT
253     ENDIF
254     SET COLOR TO R+/E, R+/B
255     @ 20,62 GET ACCEPT PICT "!"
256     READ
257 ENDDO
258 SET COLOR TO W/B, W/B
259 @ 20,10 SAY SPACE(55)
260 *
261 *     IF ENTRIES ARE CORRECT, ADD THEM TO DATABASE.
262 *
263     IF ACCEPT = "Y"
264         APPEND BLANK
265         REPLACE SITENO      WITH MSITE
266         REPLACE FEATURENO   WITH MFEAT
267         REPLACE MANLDESC    WITH MMANDESC
268     ENDIF
269 *
270 ENDIF
271 *
272 SET COLOR TO R+/B, R+/B
273 STORE "C" TO CHOICE
274 @ 22,58 GET CHOICE PICT "!"
275 READ
276 *
277 *     ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
278 *
279     DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
280         IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
281             SET COLOR TO W+/R, W+/R
282             @ 24,24 SAY " Response must be either C or X "
283             DO DELAY
284             STORE "C" TO CHOICE
285         ENDIF
286         SET COLOR TO R+/B, R+/B
287         @ 22,58 GET CHOICE PICT "!"
288         READ
289     ENDDO
290 *
291 *     SKIP TO THE NEXT RECORD TO BE REVIEWED
292 *
293     IF CHOICE = "C" THEN
294         STORE " " TO MCLIN
295         STORE SPACE(30) TO MDESCRIPT
296         STORE SPACE(26) TO MMANDESC
297         SET COLOR TO /BR, /BR
298         @ 12,45 SAY " "
299         @ 13,45 SAY MCLIN PICT "9999"
300         @ 14,45 SAY MDESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"

```

```
301      a 17,45 SAY MMANDESC PICT "!!!!!!!!!!!!!!!!!!!!!!!"
302      SKIP
303  ENDIF
304  *
305  * USER HAS DECIDED TO EXIT THE REVIEW
306  *
307      IF CHOICE = "X"
308          EXIT
309      ENDIF
310  *
311  ENDDO WHILE .T.
312  *
313  * RETURN TO CALLING PROGRAM.
314  *
315  RELEASE ALL LIKE M*, ACCEPT, CHOICE, ERROR, FEATURES, GETOUT,;
316      NOFIND, NOSITE, SITES
317  CLOSE DATABASES
318  RETURN
319  *****
```

Page 1

MANULCMD.PRG Program Listing

```

1  * PROCEDURE MANULCMD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              LCDR WINSTON H. BUCKLEY, SC, USN
5  *              LCDR ROBERT F. BRADO, USN
6  *              LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER THE OPPORTUNITY TO ADD A MANUAL
9  *              RECORD, UPDATE AN EXISTING RECORD, DELETE AN EXISTING
10 *              RECORD OR REVIEW CURRENT RECORDS.
11 *
12 * INPUT FILES   : NONE.
13 *
14 * OUTPUT FILES  : NONE.
15 *
16 * CALLED BY     : MAINMENU.PRG
17 *
18 * MODULES CALLED : MANULADD.PRG, MANULUPD.PRG, MANULDEL.PRG,
19 *              MANULREV.PRG
20 *
21 * LOCAL VARIABLES: SELEKT
22 *
23 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
24 *
25 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE SELECTION.
26 *
27 STORE "1" TO SELEKT
28 DO WHILE SELEKT < "5"
29     SET COLOR TO W/B, W/B, B
30     CLEAR
31     ?? FLASH + "W.MANULCMD/"
32     SET CONSOLE OFF
33     WAIT TO SELEKT
34     SET CONSOLE ON
35 *
36 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
37 *
38 DO CASE
39 *
40 *     CALL THE MANUAL ADD PROGRAM.
41 *     CASE SELEKT = "1"
42 *         DO MANULADD
43 *
44 *     CALL THE MANUAL UPDATE PROGRAM.
45 *     CASE SELEKT = "2"
46 *         DO MANULUPD
47 *
48 *     CALL MANUAL DELETION PROGRAM.
49 *     CASE SELEKT = "3"
50 *         DO MANULDEL

```

Page 2

MANULCMD.PRG Program Listing

```
51 | *
52 | *      CALL MANUAL REVIEW PROGRAM.
53 | *      CASE SELEKT = "4"
54 | *      DO MANULREV
55 | *
56 | *      RETURN TO THE MAIN MENU PROGRAM.
57 | *      CASE SELEKT = "5"
58 | *
59 | *      ENDCASE
60 | *
61 | *      ENDDO (WHILE SELEKT < "5")
62 | *
63 | *      RETURN TO THE CALLING PROGRAM
64 | *
65 | *      RETURN
66 | *****
```

Page 1

MANULDEL.PRG Program Listing

```

1  * PROCEDURE MANULDEL.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              LCDR WINSTON H. BUCKLEY, SC, USN
5  *              LCDR ROBERT F. BRADO, USN
6  *              LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : DELETE MANUAL RECORDS FROM THE MANUAL DATABASE FILE.
9  *
10 * INPUT FILES  : MANUAL.DBF, MANULSIT.NDX
11 *
12 * CALLED BY    : MANULCMD.PRG
13 *
14 * MODULES CALLED : DELAY.PRG
15 *
16 * GLOBAL VARIABLE: HIFNUM, HISITE, LOFNUM, LOSITE
17 *
18 * LOCAL VARIABLES: ACCEPT, CHOICE, ERROR, FEATURES, FIRST_REC,
19 *                 LAST_REC, MCLIN, MDESCIPT, MESSAGE, MFEAT,
20 *                 MKEY, MMANDESC, MSITE, PACKEM, SITES
21 *
22 * DATE LAST TIME MODIFIED =====> 24 DECEMBER 1985 <=====
23 *
24 * CASE SELECTION = 3      DELETE AN EXISTING MANUAL RECORD
25 *
26 SET DELETED ON
27 SET ESCAPE OFF
28 SET TALK OFF
29 USE MANUAL
30 GO TOP
31 SET COLOR TO W+/B, W+/B, B
32 CLEAR
33 IF EOF() = .T. THEN
34     SET COLOR TO W+/R, W+/R
35     @ 13,25 SAY " The MANUALS Database is EMPTY! "
36     DO DELAY
37     RETURN
38 ENDIF
39 SELECT 1
40 USE MANUAL INDEX MANULSIT
41 GO BOTTOM
42 STORE RECNO() TO LAST_REC
43 *
44 ?? FLASH + "S.MANUALS.SCR/"
45 @ 24,0 SAY SPACE(80)
46 @ 22,10 SAY SPACE(60)
47 SET COLOR TO R+/ , R+/
48 @ 3,26 SAY " MANUAL DELETION FORMAT "
49 SET COLOR TO W+/B, W+/B
50 @ 22,23 SAY "Enter C to continue or X to exit:"

```

Page 2

MANULDEL.PRG Program Listing

```
51 SET COLOR TO R+/B, R+/B
52 @ 22,29 SAY "C"
53 @ 22,46 SAY "X"
54 STORE SPACE(9) + "Enter the Site Number for the Manual " +;
55   "Description to be Deleted" + SPACE(9) TO SITES
56 STORE SPACE(10) + "Enter the Feature Number for the Manual " +;
57   "Description Deletion" + SPACE(10) TO FEATURES
58 STORE "Records marked for deletion have been deleted and " +;
59   "CAN NOT be recovered" TO PACKEM
60 STORE "Are you sure you want to delete this description? " +;
61   "(Yes or No):" TO MESSAGE
62 *
63 SET COLOR TO /W, /W
64 @ 24,0 SAY SITES
65 *
66 * ENSURE THAT THE SITE NUMBER IS A VALID SITE
67 *
68 STORE ' ' TO MSITE
69 DO WHILE .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
70   SET COLOR TO /BR, /BR
71   STORE LOSITE TO MSITE
72   @ 9,45 GET MSITE PICT '99'
73   READ
74   IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
75     SET COLOR TO W/B, W/B
76     @ 24,0 SAY SPACE(80)
77     SET COLOR TO W+/R, W+/R
78     STORE ' Response must be between ' + LOSITE +;
79       ' and ' + HISITE + ' ' TO ERROR
80     @ 24,22 SAY ERROR
81     DO DELAY
82     SET COLOR TO /W, /W
83     @ 24,0 SAY SITES
84     LOOP
85   ELSE
86     GO TOP
87     FIND &MSITE
88     IF EOF() = .T. THEN
89       SET COLOR TO W/B, W/B
90       @ 24,0 SAY SPACE(80)
91       SET COLOR TO W+/R, W+/R
92       STORE ' No record for site number ' + MSITE +;
93         ' exists, try again ' TO ERROR
94       @ 24,16 SAY ERROR
95       DO DELAY
96       SET COLOR TO /W, /W
97       @ 24,0 SAY SITES
98       STORE '99' TO MSITE
99     ENDIF EOF() = .T.
100  ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
```



```
101 ENDDO WHILE .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
102 *
103 SET COLOR TO W/B, W/B
104 @ 24,0 SAY SPACE(80)
105 STORE " " TO MFEAT
106 SET COLOR TO /W, /W
107 @ 24,0 SAY FEATURES
108 *
109 * ENSURE THAT THE FEATURE IS A VALID FEATURE
110 *
111 DO WHILE .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
112 SET COLOR TO /BR, /BR
113 STORE LOFNUM TO MFEAT
114 @ 12,45 GET MFEAT PICT '999999'
115 READ
116 IF .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
117 SET COLOR TO W/B, W/B
118 @ 24,0 SAY SPACE(80)
119 SET COLOR TO W+/R, W+/R
120 STORE ' Response must be between ' + LOFNUM + ;
121 ' and ' + HIFNUM + ' ' TO ERROR
122 @ 24,18 SAY ERROR
123 DO DELAY
124 SET COLOR TO /W, /W
125 @ 24,0 SAY FEATURES
126 LOOP
127 ELSE
128 STORE MSITE + MFEAT TO MKEY
129 GO TOP
130 FIND &MKEY
131 IF EOF() = .T. THEN
132 SET COLOR TO W/B, W/B
133 @ 24,0 SAY SPACE(80)
134 SET COLOR TO W+/R, W+/R
135 STORE ' No record exists for feature number ' + MFEAT + ;
136 ', try again ' TO ERROR
137 @ 24,12 SAY ERROR
138 DO DELAY
139 SET COLOR TO /W, /W
140 @ 24,0 SAY FEATURES
141 STORE '999999' TO MFEAT
142 ENDIF EOF() = .T.
143 ENDIF .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
144 ENDDO WHILE .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
145 *
146 SET COLOR TO W/B, W/B
147 @ 24,0 SAY SPACE(80)
148 *
149 DO WHILE .T.
150 SET COLOR TO R+/B, R+/B
```

Page 4

MANULDEL.PRG Program Listing

```
151 @ 6,47 SAY RECNO() PICT "9999"
152 STORE FEATURENO TO MFEAT
153 SELECT 2
154 USE DESCRIP INDEX DESCRIP.NDX
155 FIND &MFEAT
156 STORE CLIN TO MCLIN
157 STORE DESCRIPT TO MDESCRIPT
158 SELECT 1
159 SET COLOR TO /BR, /BR
160 @ 9,45 SAY SITENO PICT '99'
161 @ 12,45 SAY FEATURENO PICT '999999'
162 @ 13,45 SAY MCLIN PICT "9999"
163 @ 14,45 SAY MDESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
164 @ 17,45 SAY MANLDESC PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
165 SET COLOR TO W/B, W/B
166 @ 24,0 SAY SPACE(80)
167 *
168 * ASK THE USER IF HE/SHE IS SURE ABOUT THE DELETION
169 *
170 SET COLOR TO W+/B, W+/B
171 @ 20,06 SAY MESSAGE
172 SET COLOR TO R+/B, R+/B
173 @ 20,58 SAY "Y"
174 @ 20,65 SAY "N"
175 STORE "N" TO ACCEPT
176 @ 20,70 GET ACCEPT PICT "!"
177 READ
178 *
179 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
180 *
181 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
182 IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
183 SET COLOR TO W+/R, W+/R
184 @ 24,24 SAY " Response must be either N or Y "
185 DO DELAY
186 STORE "N" TO ACCEPT
187 ENDIF
188 SET COLOR TO R+/B, R+/B
189 @ 20,70 GET ACCEPT PICT "!"
190 READ
191 ENDDO
192 SET COLOR TO W/B, W/B
193 @ 20,05 SAY SPACE(70)
194 *
195 * IF ENTRIES ARE CORRECT, DELETE THEM FROM THE DATABASE,
196 * IF NOT RECOVER THEM
197 *
198 IF ACCEPT = "Y"
199 DELETE
200 ENDIF
```

```
201 *
202 SET COLOR TO R+/B, R+/B
203 STORE "C" TO CHOICE
204 @ 22,58 GET CHOICE PICT "!"
205 READ
206 *
207 * ENSURE THAT THE USER'S RESPONSE IS EITHER "C" OR "X"
208 *
209 DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
210 IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
211 SET COLOR TO W+/R, W+/R
212 @ 24,24 SAY " Response must be either C or X "
213 DO DELAY
214 STORE "C" TO CHOICE
215 ENDIF
216 SET COLOR TO R+/B, R+/B
217 @ 22,58 GET CHOICE PICT "!"
218 READ
219 ENDDO
220 *
221 * SKIP TO THE NEXT RECORD TO BE REVIEWED
222 *
223 IF CHOICE = "C" THEN
224 IF RECNO() = LAST_REC THEN
225 GO TOP
226 ELSE
227 SKIP
228 ENDIF
229 ENDIF
230 *
231 * USER HAS DECIDED TO EXIT THE REVIEW
232 *
233 IF CHOICE = "X"
234 SET COLOR TO W+/R, W+/R
235 @ 24,0
236 @ 24,6 SAY PACKEM
237 SET COLOR TO W/B, W/B
238 PACK
239 EXIT
240 ENDIF
241 *
242 ENDDO WHILE .T.
243 *
244 * RETURN TO CALLING PROGRAM.
245 *
246 RELEASE ALL LIKE M*, ACCEPT, CHOICE, ERROR, FEATURES, FIRST_REC,;
247 LAST_REC, PACKEM, SITES
248 CLOSE DATABASES
249 RETURN
250 *****
```

Page 1

MANULREV.PRG Program Listing

```

1 * PROCEDURE MANULREV.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              LCDR WINSTON H. BUCKLEY, SC, USN
5 *              LCDR ROBERT F. BRADO, USN
6 *              LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : TO ENABLE THE USER TO REVIEW ALL THE RECORDS IN
9 *              THE MANUAL DATABASE
10 *
11 * INPUT FILES  : MANUAL.DBF, MANULSIT.NDX
12 *
13 * CALLED BY    : MANULCMD.PRG
14 *
15 * MODULES CALLED : DELAY.PRG
16 *
17 * GLOBAL VARIABLE: HIFNUM, HISITE, LOFNUM, LOSITE
18 *
19 * LOCAL VARIABLES: ACCEPT, CHOICE, CURRENTNO, EOF, ERROR, FIRST_REC,
20 *                 LAST_REC, MCLIN, MDESCRIPT, MFEAT, MSITE, TOF
21 *
22 * DATE LAST TIME MODIFIED =====> 26 DECEMBER 1985 <=====
23 *
24 * CASE SELECTION = 4      REVIEW EXISTING MANUAL RECORDS
25 *
26 SET ESCAPE OFF
27 SET TALK OFF
28 USE MANUAL
29 GO TOP
30 SET COLOR TO W+/B, W+/B, B
31 CLEAR
32 IF EOF() = .T. THEN
33   SET COLOR TO W+/R, W+/R
34   @ 13,25 SAY " The MANUALS Database is EMPTY! "
35   DO DELAY
36   RETURN
37 ENDIF
38 ?? FLASH + "S.MANUALS.SCR/"
39 @ 24,0 SAY SPACE(80)
40 SET COLOR TO R+/ , R+/
41 @ 3,26 SAY " MANUAL REVIEW FORMAT "
42 SELECT 1
43 STORE " Enter 00 to start at TOF, 99 to start at EOF or a site number " +;
44   "between " + LOSITE + " and " + HISITE + " " TO MESSAGE
45 SET COLOR TO /W, /W
46 @ 24,0 SAY MESSAGE
47 STORE '88' TO MSITE
48 DO WHILE .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
49   SET COLOR TO /BR , /BR
50   STORE '00' TO MSITE

```

```

51 @ 09,45 GET MSITE PICT '99'
52 READ
53 IF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99') THEN
54     SET COLOR TO W/B, W/B
55     @ 24,0 SAY SPACE(80)
56     SET COLOR TO W+/R, W+/R
57     STORE 'Response must be between ' + LOSITE + ' and ' +
58         HISITE + ', Zero (00) or 99 ' TO ERROR
59     @ 24,13 SAY ERROR
60     DO DELAY
61     SET COLOR TO /W, /W
62     @ 24,0 SAY MESSAGE
63     LOOP
64 ELSE
65     IF (MSITE = '00' .OR. MSITE = '99') THEN
66         USE MANUAL
67         IF MSITE = '00' THEN
68             GO BOTTOM
69             STORE RECNO() TO LAST_REC
70             GO TOP
71             STORE RECNO() TO FIRST_REC
72         ELSE
73             GO TOP
74             STORE RECNO() TO FIRST_REC
75             GO BOTTOM
76             STORE RECNO() TO LAST_REC
77         ENDIF MSITE = '00'
78         EXIT
79     ELSE
80         USE MANUAL INDEX MANULSIT
81         GO TOP
82         FIND &MSITE
83         IF EOF() = .T. THEN
84             SET COLOR TO W/B, W/B
85             @ 24,0 SAY SPACE(80)
86             SET COLOR TO W+/R, W+/R
87             STORE "No records exist for site number " + MSITE +
88                 ", try again " TO ERROR
89             @ 24,16 SAY ERROR
90             DO DELAY
91             SET COLOR TO /W, /W
92             @ 24,0 SAY MESSAGE
93             STORE '88' TO MSITE
94             LOOP
95         ELSE
96             EXIT
97         ENDIF
98     ENDIF
99 ENDIF
100 ENDDO WHILE

```

```

101 *
102 STORE SPACE(10) + 'Enter "00    " to start at TOF or a six digit ' +;
103     'feature number' + SPACE(10) TO MESSAGE
104 IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
105     SET COLOR TO /W, /W
106     @ 24,0 SAY MESSAGE
107     DO WHILE .T.
108         SET COLOR TO /BR, /BR
109         STORE '00    ' TO MFEAT
110         @ 12,45 GET MFEAT PICT '999999'
111         READ
112         IF .NOT. ((MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) .OR.;
113             MFEAT = '00    ') THEN
114             SET COLOR TO W/B, W/B
115             @ 24,0 SAY SPACE(80)
116             SET COLOR TO W+/R, W+/R
117             STORE ' Response must be between ' + LOFNUM + ' and ' +;
118                 HIFNUM + ' or Zero (00) ' TO ERROR
119             @ 24,9 SAY ERROR
120             DO DELAY
121             SET COLOR TO /W, /W
122             @ 24,0 SAY MESSAGE
123             LOOP
124     ELSE
125         IF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) THEN
126             IF MFEAT = '99    ' THEN
127                 SET COLOR TO W/B, W/B
128                 @ 24,0 SAY SPACE(80)
129                 SET COLOR TO W+/R, W+/R
130                 STORE ' Response must be between ' + LOFNUM +;
131                     ' and ' + HIFNUM + ' or Zero (00) ' TO ERROR
132                 @ 24,9 SAY ERROR
133                 DO DELAY
134                 SET COLOR TO /W, /W
135                 @ 24,0 SAY MESSAGE
136                 LOOP
137             ENDIF MFEAT = '99    '
138             STORE MSITE + MFEAT TO MKEY
139             USE MANUAL INDEX MANULSIT
140             GO TOP
141             FIND &MKEY
142             IF EOF() = .T. THEN
143                 SET COLOR TO W/B, W/B
144                 @ 24,0 SAY SPACE(80)
145                 SET COLOR TO W+/R, W+/R
146                 STORE " No record with feature number " + MFEAT +;
147                     " exists, try again " TO ERROR
148                 @ 24,12 SAY ERROR
149                 DO DELAY
150                 SET COLOR TO /W, /W

```

```

151             @ 24,0 SAY MESSAGE
152             LOOP
153             ELSE
154             EXIT
155             ENDIF EOF() = .T.
156             ELSE
157             GO TOP
158             EXIT
159             ENDIF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
160             ENDIF
161             ENDDO WHILE .T.
162             ENDIF .NOT. (MSITE >= LOSITE .OR. MSITE <= HISITE)
163             *
164             SET COLOR TO W/B, W/B
165             @ 24,0 SAY SPACE(80)
166             STORE " At beginning of records for site number " +;
167             MSITE + " " TO TOF
168             STORE " At end of records for site number " + MSITE + " " TO EOF
169             DO WHILE .T.
170             SET COLOR TO R+/B, R+/B
171             @ 6,47 SAY RECNO() PICT "9999"
172             *
173             * CLEAR SCREEN AND SET INITIAL VALUES FOR VARIABLES TO BE
174             * ADDED TO THE FILE. THE M PREFIX INDICATES MEMORY VARIABLES
175             * DISTINGUISHING THEM FROM THEIR CORRESPONDING DATABASE FIELDS.
176             *
177             *
178             STORE FEATURENO TO MFEAT
179             SELECT 2
180             USE DESCRIP INDEX DESCRIP
181             FIND &MFEAT
182             STORE CLIN TO MCLIN
183             STORE DESCRIPT TO MDESCIPT
184             SELECT 1
185             SET COLOR TO /BR, /BR
186             @ 09,45 SAY SITENO PICT "99"
187             @ 12,45 SAY FEATURENO PICT "999999"
188             @ 13,45 SAY MCLIN PICT "9999"
189             @ 14,45 SAY MDESCIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
190             @ 17,45 SAY MANLDESC PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
191             *
192             SET COLOR TO R+/B, R+/B
193             STORE "N" TO CHOICE
194             @ 22,67 GET CHOICE PICT "!"
195             READ
196             *
197             * ENSURE THAT THE USER'S RESPONSE IS EITHER "N", "P" OR "X"
198             *
199             DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
200             IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN

```

Page 5

MANULREV.PRG Program Listing

```
201          SET COLOR TO W+/R,W+/R
202          @ 24,22 SAY " Response must be either N, P or X "
203          DO DELAY
204          STORE "N" TO CHOICE
205      ENDIF
206      SET COLOR TO R+/B,R+/B
207      @ 22,67 GET CHOICE PICT "!"
208      READ
209      ENDDO
210  *
211  * SKIP TO THE NEXT RECORD TO BE REVIEWED
212  *
213      IF CHOICE = "N" THEN
214          IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
215              SKIP
216              IF EOF() = .T. THEN
217                  SKIP - 1
218                  SET COLOR TO W+/R, W+/R
219                  @ 24,21 SAY EOF
220                  DO DELAY
221              ELSE
222                  IF .NOT. (SITENO = MSITE) THEN
223                      SKIP - 1
224                      SET COLOR TO W+/R, W+/R
225                      @ 24,21 SAY EOF
226                      DO DELAY
227                  ENDIF
228                  ENDF EOF() = .T.
229              ELSE
230                  IF RECNO() = LAST_REC THEN
231                      GO TOP
232                  ELSE
233                      SKIP
234                  ENDIF
235              ENDF (MSITE >= LOSITE .AND. MSITE <= HISITE)
236          ENDF CHOICE = "N"
237  *
238  * SKIP TO THE PREVIOUS RECORD
239  *
240      IF CHOICE = "P" THEN
241          STORE RECNO() TO CURRENINO
242          IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
243              SKIP - 1
244              IF BOF() = .T. THEN
245                  GOTO CURRENINO
246                  SET COLOR TO W+/R, W+/R
247                  @ 24,16 SAY TOF
248                  DO DELAY
249              ELSE
250                  IF .NOT. (SITENO = MSITE) THEN
```



```

251             SKIP
252             SET COLOR TO W+/R, W+/R
253             @ 24,16 SAY TOF
254             DO DELAY
255             ENDIF
256             ENDIF BOF() = .T.
257         ELSE
258             IF RECNO() = FIRST_REC THEN
259                 GO BOTTOM
260             ELSE
261                 SKIP - 1
262             ENDIF
263             ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
264             ENDIF CHOICE = "P"
265         *
266         * USER HAS DECIDED TO EXIT THE REVIEW
267         *
268         IF CHOICE = "X"
269             EXIT
270         ENDIF
271     *
272     ENDDO WHILE .T.
273     *
274     * RETURN TO CALLING PROGRAM.
275     *
276     RELEASE ALL LIKE M*, ACCEPT, CHOICE, CURRENTNO, EOF, FIRST_REC, LAST_REC, TOF
277     CLOSE DATABASES
278     RETURN
279     *****

```

Page 1

MANULUPD.PRG Program Listing

```

1  * PROCEDURE MANULUPD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : TO ENABLE THE USER TO UPDATE SELECTED RECORDS IN
9  *               THE MANUAL DATABASE
10 *
11 * INPUT FILES   : MANUAL.DBF, MANULSIT.NDX
12 *
13 * CALLED BY     : MANULCMD.PRG
14 *
15 * MODULES CALLED : DELAY.PRG
16 *
17 * GLOBAL VARIABLE: HIFNUM, HISITE, LOFNUM LOSITE
18 *
19 * LOCAL VARIABLES: ACCEPT, ANS, CHOICE, EOF, ERROR, MCLIN, MDATE,
20 *                 MDESCIPT, MFEAT, MSITE, TOF
21 *
22 * DATE LAST TIME MODIFIED =====> 26 DECEMBER 1985 <=====
23 *
24 * CASE SELECTION = 2      UPDATE AN EXISTING MANUAL DESCRIPTION
25 *
26 SET ESCAPE OFF
27 SET TALK OFF
28 USE MANUAL
29 GO TOP
30 SET COLOR TO W+/B, W+/B, B
31 CLEAR
32 IF EOF() = .T. THEN
33     SET COLOR TO W+/R, W+/R
34     @ 13,25 SAY " The MANUALS Database is EMPTY! "
35     DO DELAY
36     RETURN
37 ENDIF
38 ?? FLASH + "S.MANUALS.SCR/"
39 @ 24,0 SAY SPACE(80)
40 SET COLOR TO R+/ , R+/
41 @ 3,26 SAY " MANUAL UPDATE FORMAT "
42 SELECT 1
43 STORE " Enter 00 to start at TOF, 99 to start at EOF or a site number " +
44     "between " + LOSITE + " and " + HISITE + " " TO MESSAGE
45 SET COLOR TO /W, /W
46 @ 24,0 SAY MESSAGE
47 STORE '88' TO MSITE
48 DO WHILE .NOT. ((MSITE = '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
49     SET COLOR TO /BR , /BR
50     STORE '00' TO MSITE

```

```

51 @ 09,45 GET MSITE PICT '99'
52 READ
53 IF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
54     SET COLOR TO W/B, W/B
55     @ 24,0 SAY SPACE(80)
56     SET COLOR TO W+/R, W+/R
57     STORE ' Response must be between ' + LOSITE + ' and ' +;
58         HISITE + ', Zero (00) or 99 ' TO ERROR
59     @ 24,13 SAY ERROR
60     DO DELAY
61     SET COLOR TO /W, /W
62     @ 24,0 SAY MESSAGE
63     LOOP
64 ELSE
65     IF (MSITE = '00' .OR. MSITE = '99') THEN
66         USE MANUAL
67         IF MSITE = '00' THEN
68             GO BOTTOM
69             STORE RECNO() TO LAST_REC
70             GO TOP
71             STORE RECNO() TO FIRST_REC
72         ELSE
73             GO TOP
74             STORE RECNO() TO FIRST_REC
75             GO BOTTOM
76             STORE RECNO() TO LAST_REC
77         ENDIF MSITE = '00'
78         EXIT
79     ELSE
80         USE MANUAL INDEX MANULSIT
81         GO TOP
82         FIND &MSITE
83         IF EOF() = .T. THEN
84             SET COLOR TO W/B, W/B
85             @ 24,0 SAY SPACE(80)
86             SET COLOR TO W+/R, W+/R
87             STORE " No records exist for site number " + MSITE +;
88                 ", try again " TO ERROR
89             @ 24,16 SAY ERROR
90             DO DELAY
91             SET COLOR TO /W, /W
92             @ 24,0 SAY MESSAGE
93             STORE '88' TO MSITE
94             LOOP
95         ENDIF EOF() = .T.
96     ENDIF (MSITE = '00' .OR. MSITE = '99')
97 ENDIF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
98 ENDDO WHILE .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')
99 *
100 STORE " At beginning of records for site number " +;

```

```

101 MSITE + " " TO TOF
102 STORE " At end of records for site number " + MSITE + " " TO EOF
103 STORE ' Enter "00      " to start at TOF or a six digit feature' +;
104     ' number ( ' + LOFNUM + ' - ' + HIFNUM + ' ) ' TO MESSAGE
105 IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
106     SET COLOR TO /W, /W
107     @ 24,0 SAY MESSAGE
108     DO WHILE .T.
109         SET COLOR TO /BR, /BR
110         STORE '00      ' TO MFEAT
111         @ 12,45 GET MFEAT PICT '999999'
112         READ
113         IF .NOT. ((MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) .OR.;
114             MFEAT = '00      ') THEN
115             SET COLOR TO W/B, W/B
116             @ 24,0 SAY SPACE(80)
117             SET COLOR TO W+/R, W+/R
118             STORE ' Response must be between ' + LOFNUM + ' and ' +;
119                 HIFNUM + ' or Zero (00) ' TO ERROR
120             @ 24,9 SAY ERROR
121             DO DELAY
122             SET COLOR TO /W, /W
123             @ 24,0 SAY MESSAGE
124             LOOP
125     ELSE
126         IF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) THEN
127             IF MFEAT = '99      ' THEN
128                 SET COLOR TO W/B, W/B
129                 @ 24,0 SAY SPACE(80)
130                 SET COLOR TO W+/R, W+/R
131                 STORE ' Response must be between ' + LOFNUM +;
132                     'and ' + HIFNUM + ' or Zero (00) ' TO ERROR
133                 @ 24,9 SAY ERROR
134                 DO DELAY
135                 SET COLOR TO /W, /W
136                 @ 24,0 SAY MESSAGE
137                 LOOP
138             ENDIF MFEAT = '99      '
139             STORE MSITE + MFEAT TO MKEY
140             USE MANUAL INDEX MANULSIT
141             GO TOP
142             FIND &MKEY
143             IF EOF() = .T. THEN
144                 SET COLOR TO W/B, W/B
145                 @ 24,0 SAY SPACE(80)
146                 SET COLOR TO W+/R, W+/R
147                 STORE " No record with feature number " + MFEAT +;
148                     " exists, try again " TO ERROR
149                 @ 24,12 SAY ERROR
150                 DO DELAY

```

```

151             SET COLOR TO /W, /W
152             @ 24,0 SAY MESSAGE
153             LOOP
154             ELSE
155             EXIT
156             ENDIF EOF() = .T.
157             ELSE
158             GO TOP
159             EXIT
160             ENDIF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
161             ENDIF
162             ENDDO WHILE .T.
163             ENDIF .NOT. (MSITE >= IOSITE .OR. MSITE <= HISITE)
164             *
165             STORE SPACE(16) + 'Press "Page Down" key to terminate record update' +;
166             SPACE(16) TO MESSAGE
167             STORE 1 TO INTRO
168             DO WHILE .T.
169             *
170             *   INFORM THE USER OF HOW TO TERMINATE THE UPDATE OF A RECORD
171             *
172             IF INTRO = 1 THEN
173             STORE 0 TO INTRO
174             ?? FLASH + 'W.MANULUPD/'
175             SET CONSOLE OFF
176             WAIT TO ANS
177             SET CONSOLE ON
178             ENDIF
179             *
180             SET COLOR TO R+/B, R+/B
181             @ 6,47 SAY RECNO() PICT "9999"
182             SET COLOR TO /W, /W
183             @ 24,0 SAY MESSAGE
184             *
185             *   STORING THE OLD RECORD TO A WORK RECORD AREA. THE M PREFIX
186             *   INDICATES MEMORY VARIABLES DISTINGUISHING THEM FROM THEIR
187             *   CORRESPONDING DATABASE FIELDS.
188             *
189             *
190             STORE FEATURENO TO MFEAT
191             STORE MANLDESC TO MMANDESC
192             *
193             SELECT 2
194             USE DESCRIP INDEX DESCRIP
195             FIND &MFEAT
196             STORE CLIN TO MCLIN
197             STORE DESCPT TO MDESCPT
198             SELECT 1
199             SET COLOR TO /BR, /BR
200             @ 09,45 SAY SITENO PICT "99"

```

```

201 @ 12,45 SAY FEATURENO PICT "999999"
202 @ 13,45 SAY MCLIN PICT "9999"
203 @ 14,45 SAY MDESCIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
204 @ 17,45 GET MMANDESC PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
205 READ
206 SET COLOR TO W/B, W/B
207 @ 24,0 SAY SPACE(80)
208 *
209 IF .NOT. (MANLDESC = MMANDESC) THEN
210 *
211 *     ASK THE USER IF HE/SHE DESIRES TO ACCEPT THE CHANGES.
212 *
213     SET COLOR TO W+/B, W+/B
214     @20,12 SAY "Do you want to accept the changes? (Yes or No):"
215     SET COLOR TO R+/B, R+/B
216     @20,49 SAY "Y"
217     @20,56 SAY "N"
218     STORE "N" TO ACCEPT
219     @20,62 GET ACCEPT PICT "!"
220     READ
221 *
222 *     ENSURE THAT THE USER'S PROMPT IS EITHER "Y" OR "N"
223 *
224     DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
225         IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
226             SET COLOR TO W+/R, W+/R
227             @ 24,24 SAY " Response must be either N or Y "
228             DO DELAY
229             STORE "N" TO ACCEPT
230         ENDIF
231         SET COLOR TO R+/B, R+/B
232         @20,62 GET ACCEPT PICT "!"
233         READ
234     ENDDO
235     SET COLOR TO W/B, W/B
236     @ 20,10 SAY SPACE(60)
237 *
238     IF ACCEPT = "Y" THEN
239         REPLACE MANLDESC WITH MMANDESC
240     ELSE
241         SET COLOR TO /BR, /BR
242         @ 17,45 SAY MANLDESC PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
243     ENDIF
244 ENDIF
245 *
246 *
247     SET COLOR TO R+/B, R+/B
248     STORE "N" TO CHOICE
249     @ 22,67 GET CHOICE PICT "!"
250     READ

```

```

251  *
252  *   ENSURE THAT THE USER'S RESPONSE IS EITHER "N", "P" OR "X"
253  *
254  DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
255      IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
256          SET COLOR TO W+/R, W+/R
257          @ 24,22 SAY " Response must be either N, P or X "
258          DO DELAY
259          STORE "N" TO CHOICE
260      ENDIF
261      SET COLOR TO R+/B, R+/B
262      @ 22,67 GET CHOICE PICT "!"
263      READ
264  ENDDO
265  *
266  *   SKIP TO THE NEXT RECORD TO BE REVIEWED
267  *
268  IF CHOICE = "N" THEN
269      IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
270          SKIP
271          IF EOF() = .T. THEN
272              SKIP - 1
273              SET COLOR TO W+/R, W+/R
274              @ 24,21 SAY EOF
275              DO DELAY
276          ELSE
277              IF .NOT. (SITENO = MSITE) THEN
278                  SKIP - 1
279                  SET COLOR TO W+/R, W+/R
280                  @ 24,21 SAY EOF
281                  DO DELAY
282              ENDIF
283              ENDIF EOF() = .T.
284          ELSE
285              IF RECNO() = LAST_REC THEN
286                  GO TOP
287              ELSE
288                  SKIP
289              ENDIF
290          ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
291      ENDIF CHOICE = "N"
292  *
293  *   SKIP TO THE PREVIOUS RECORD
294  *
295  IF CHOICE = "P" THEN
296      STORE RECNO() TO CURRENINO
297      IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
298          SKIP - 1
299          IF BOF() = .T. THEN
300              GOTO CURRENINO

```

```

301      SET COLOR TO W+/R, W+/R
302      @ 24,16 SAY TOF
303      DO DELAY
304      ELSE
305          IF .NOT. (SITE NO = MSITE) THEN
306              SKIP
307              SET COLOR TO W+/R, W+/R
308              @ 24,16 SAY TOF
309              DO DELAY
310          ENDIF
311      ENDIF BOF() = .T.
312      ELSE
313          IF RECNO() = FIRST_REC THEN
314              GO BOTTOM
315          ELSE
316              SKIP - 1
317          ENDIF
318      ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
319      ENDIF CHOICE = "P"
320      *
321      * USER HAS DECIDED TO EXIT THE REVIEW
322      *
323      IF CHOICE = "X"
324          EXIT
325      ENDIF
326      *
327      ENDDO WHILE .T.
328      *
329      * RETURN TO CALLING PROGRAM.
330      *
331      RELEASE ALL LIKE M*, ACCEPT, ANS, CHOICE, ERROR
332      CLOSE DATABASES
333      RETURN
334      *****

```



```

1  * PROCEDURE MKLABELS.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER WITH THE CAPABILITY OF RAPIDLY
9  *               GENERATING MAILING LABELS FOR ALL OF THE SPLICE
10 *               SITES.
11 *
12 * INPUT FILES   : NONE.
13 *
14 * OUTPUT FILES  : NONE.
15 *
16 * CALLED BY     : MAINMENU.PRG
17 *
18 * MODULES CALLED : DELAY.PRG
19 *
20 * LOCAL VARIABLES: COPIES, IMAGE, INTRO, LABELS, LAST_LINE,
21 *               LINECNT, MESSAGE, SKIPONE
22 *
23 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
24 *
25 * GENERATE MAILING LABELS FOR ALL OF THE SPLICE SITES.
26 *
27 SET ESCAPE OFF
28 SET EXACT ON
29 SET TALK OFF
30 SET COLOR TO W/B, W/B, B
31 CLEAR
32 ?? FLASH + "S.MKLABELS.SCR/"
33 @ 24,0 SAY SPACE(80)
34 SET COLOR TO R+/B, R+/B
35 *
36 * OBTAIN THE NUMBER OF SETS OF LABELS TO PRINT FROM THE USER
37 *
38 STORE SPACE(5) + "Input the number of sets of labels desired" +;
39 " (Range 1 - 10) or 00 TO EXIT" + SPACE(5) TO MESSAGE
40 *
41 STORE "99" TO COPIES
42 DO WHILE .NOT. (COPIES >= "00" .AND. COPIES <= "10")
43     SET COLOR TO /W, /W
44     @ 24,0 SAY MESSAGE
45     STORE "00" TO COPIES
46     SET COLOR TO R+/B, R+/B
47     @ 6,55 GET COPIES PICT "99"
48     READ
49     IF .NOT. (COPIES >= '00' .AND. COPIES <= '10')
50         SET COLOR TO W/B, W/B

```

Page 2

MKLABELS.PRG Program Listing

```
51      @ 24,0 SAY SPACE(80)
52      SET COLOR TO W+/R, W+/R
53      @ 24,22 SAY " Response must be between 00 and 10 "
54      DO DELAY
55      SET COLOR TO /W, /W
56      @ 24,0 SAY MESSAGE
57      LOOP
58  ENDIF
59 ENDDO
60 SET COLOR TO W/B, W/B
61 @ 24,0 SAY SPACE(80)
62 *
63 IF COPIES = "00" THEN
64     SET EXACT OFF
65     RELEASE COPIES, MESSAGE
66     RETURN
67 ENDIF
68 *
69 * START PRINTING LABELS
70 *
71 USE CONFIG INDEX CONFIG
72 GO 'TOP
73 STORE SPACE(15) + " Performing printer alignment test for label forms" +;
74     SPACE(15) TO MESSAGE
75 STORE "Running label forms alignment print test" TO IMAGE
76 STORE "Y" TO CHOICE
77 STORE 1 TO INTRO
78 STORE 1 TO LINECNT
79 STORE LINECNT + 8 TO SKIPONE
80 *
81 * ASK THE USER IF A PRINTER ALIGNMENT TEST IS DESIRED
82 *
83 SET COLOR TO W+/B, W+/B
84 @ 22,9 SAY "Do you desire to run a printer alignment test? (Yes or No): "
85 SET COLOR TO R+/B, R+/B
86 @ 22,57 SAY "Y"
87 @ 22,64 SAY "N"
88 DO WHILE CHOICE = "Y"
89     SET COLOR TO R+/B, R+/B
90     @ 22,70 GET CHOICE PICT "!"
91     READ
92 *
93 * ENSURE THAT THE USER'S PROMPT IS EITHER "Y" OR "N"
94 *
95 DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "Y")
96     IF .NOT. (CHOICE = "N" .OR. CHOICE = "Y") THEN
97         SET COLOR TO W+/R, W+/R
98         @ 24,24 SAY " Response must be either N or Y "
99         DO DELAY
100        STORE "Y" TO CHOICE
```

```
101      ENDIF .NOT. (CHOICE = "N" .OR. CHOICE = "Y")
102      SET COLOR TO R+/B ,R+/B
103      @ 22,70 GET CHOICE PICT "!"
104      READ
105  ENDDO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "Y")
106  *
107  IF CHOICE = "Y"
108      @ 22,70 SAY " "
109  *
110  *      DISPLAY PRINTER INFORMATION WINDOW TO USER
111  *
112      IF INTRO = 1 THEN
113          STORE 0 TO INTRO
114          ?? FLASH + "W.LABELS/"
115          SET CONSOLE OFF
116          WAIT TO ANS
117          SET CONSOLE ON
118      ENDIF INTRO = 1
119      SET COLOR TO /W, /W
120      @ 24,0 SAY MESSAGE
121      SET COLOR TO /BR, /BR
122      @ 14,19 SAY IMAGE
123      @ 15,19 SAY IMAGE
124      @ 16,19 SAY IMAGE
125      @ 17,19 SAY IMAGE
126      @ 18,19 SAY IMAGE
127      @ 19,19 SAY IMAGE
128      SET DEVICE TO PRINT
129      DO WHILE LINECNT < SKIPONE
130          @ LINECNT,1 SAY IMAGE
131          LINECNT = LINECNT + 1
132      ENDDO WHILE LINECNT < SKIPONE
133      SKIPONE = LINECNT + 8
134      SET DEVICE TO SCREEN
135      SET COLOR TO W/B, W/B
136      @ 24,0 SAY SPACE(80)
137  ELSE
138      SET COLOR TO /BR, /BR
139      @ 14,19 SAY SPACE(40)
140      @ 15,19 SAY SPACE(40)
141      @ 16,19 SAY SPACE(40)
142      @ 17,19 SAY SPACE(40)
143      @ 18,19 SAY SPACE(40)
144      @ 19,19 SAY SPACE(40)
145  LOOP
146  ENDIF CHOICE = "Y"
147  ENDDO WHILE CHOICE = "Y"
148  *
149  *      SKIP ONE BLANK LABEL PRIOR TO PRINTING SITE LABELS
150  *
```

```
151 SET DEVICE TO PRINT
152 *
153 DO WHILE LINECNT < SKIPONE
154   @ LINECNT,1 SAY " "
155   LINECNT = LINECNT + 1
156 ENDDO WHILE LINECNT < SKIPONE
157 *
158 SET DEVICE TO SCREEN
159 *
160 SET COLOR TO W+/B, W+/B
161 @ 21,10 SAY SPACE (60)
162 *
163 * DISPLAY PRINTER INFORMATION WINDOW TO USER
164 *
165 IF INTRO = 1 THEN
166   STORE 0 TO INTRO
167   ?? FLASH + "W.LABELS/"
168   SET CONSOLE OFF
169   WAIT TO ANS
170   SET CONSOLE ON
171 ENDIF INTRO = 1
172 *
173 STORE SPACE(28) + "Printing Mailing Labels " + SPACE(28) TO MESSAGE
174 SET COLOR TO /W, /W
175 @ 24,0 SAY MESSAGE
176 *
177 DO WHILE .NOT. EOF()
178   STORE TRIM(SITECITY) + ", " + TRIM(SITESTATE) + " " +
179     TRIM(SITEZIP) TO LAST_LINE
180   SET COLOR TO R+/B, R+/B
181   IF SITECO = " " THEN
182     SKIP
183     LOOP
184   ENDIF SITECO = " "
185   @ 10,46 SAY SITENO PICT "99"
186   SET COLOR TO /BR, /BR
187   @ 15,19 SAY SITECO PICT "!!!!!!!!!!!!!!!!!!!!!!"
188   @ 16,19 SAY SITENAMEFL PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
189   IF SITEADD1 > " " THEN
190     @ 17,19 SAY SITEADD1 PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
191     IF SITEADD2 > " " THEN
192       @ 18,19 SAY SITEADD2 PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
193       @ 19,19 SAY LAST_LINE
194     ELSE
195       @ 18,19 SAY LAST_LINE
196     ENDIF SITEADD2 > " "
197   ELSE
198     @ 17,19 SAY LAST_LINE
199   ENDIF SITEADD1 > " "
200 SET DEVICE TO PRINT
```

```

201 STORE 0 TO LABELS
202 DO WHILE LABELS < VAL(COPIES)
203     @ LINECNT,1 SAY SPACE(40)
204     @ LINECNT+1,1 SAY SPACE(40)
205     @ LINECNT+2,1 SAY SITECO PICT "!!!!!!!!!!!!!!!!!!!!!!"
206     @ LINECNT+3,1 SAY SITENAMEFL
207         PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
208     IF SITEADD1 > " " THEN
209         @ LINECNT+4,1 SAY SITEADD1
210         PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
211     IF SITEADD2 > " " THEN
212         @ LINECNT+5,1 SAY SITEADD2
213         PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
214         @ LINECNT+6,1 SAY LAST_LINE
215         @ LINECNT+7,1 SAY SPACE(40)
216     ELSE
217         @ LINECNT+5,1 SAY LAST_LINE
218         @ LINECNT+6,1 SAY SPACE(40)
219         @ LINECNT+7,1 SAY SPACE(40)
220     ENDIF SITEADD2 > " "
221 ELSE
222     @ LINECNT+4,1 SAY LAST_LINE
223     @ LINECNT+5,1 SAY SPACE(40)
224     @ LINECNT+6,1 SAY SPACE(40)
225     @ LINECNT+7,1 SAY SPACE(40)
226 ENDIF SITEADD1 > " "
227 LINECNT = LINECNT+8
228 IF LINECNT > 81 THEN
229     LINECNT = 1
230 ENDIF LINECNT > 81
231 LABELS = LABELS + 1
232 ENDDO WHILE LABELS < COPIES
233 SET DEVICE TO SCREEN
234 @ 14,19 SAY SPACE(40)
235 @ 15,19 SAY SPACE(40)
236 @ 16,19 SAY SPACE(40)
237 @ 17,19 SAY SPACE(40)
238 @ 18,19 SAY SPACE(40)
239 @ 19,19 SAY SPACE(40)
240 SKIP
241 ENDDO WHILE .NOT. EOF()
242 *
243 * RETURN TO THE CALLING PROGRAM
244 *
245 SET EXACT OFF
246 RELEASE COPIES, IMAGE, INTRO, LABELS, LAST_LINE, LINECNT,;
247     MESSAGE, SKIPONE
248 CLOSE DATABASES
249 RETURN
250 *****

```

Page 1

MNLSTRPT.PRG Program Listing

```

1  * PROCEDURE MNLSTRPT.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              : LCDR WINSTON H. BUCKLEY, SC, USN
5  *              : LCDR ROBERT F. BRADO, USN
6  *              : LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER A SPLICE MANUAL SITE
9  *              : LEVEL REPORT.
10 *
11 * INPUT FILES  : MANUAL.DBF, TEMPONE.DBF, DESCRIPT.DBF,
12 *              : DESCRIPT.NDX
13 *
14 * OUTPUT FILES : NONE.
15 *
16 * CALLED BY    : SITERPTS.PRG
17 *
18 * MODULES CALLED : DELAY.PRG
19 *
20 * GLOBAL VARIABLE: HISITE, LOSITE
21 *
22 * LOCAL VARIABLES: ACCEPT, CHOICE, ERROR, LINECT, MSITE, PAGENO,
23 *              : TODAY, TODATE
24 *
25 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
26 *
27 * CASE SELECTION = 2    MANUAL SITE LEVEL REPORT
28 *
29 * CREATE THE SPLICE MANUAL SITE REPORT AND CHECK IF THE REPORT
30 * IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
31 *
32 SET ESCAPE OFF
33 SET TALK OFF
34 SET COLOR TO W+/B, W+/B, B
35 CLEAR
36 USE MANUAL
37 GO TOP
38 IF EOF() = .T. THEN
39     SET COLOR TO W+/R, W+/R
40     @ 13,25 SAY " The MANUALS Database is EMPTY! "
41     DO DELAY
42     RETURN
43 ENDIF
44 ?? FLASH + "S.REPORTS.SCR/"
45 @ 24,0 SAY SPACE(80)
46 SET COLOR TO R+/ , R+/
47 @ 2,27 SAY " SITE LEVEL MANUAL REPORT "
48 *
49 * ENSURE THAT TEMPORARY DATABASE AND INDEX DO NO EXIST, IF SO ERASE THEM
50 *

```

```
51 SET CONSOLE OFF
52 ERASE TEMPONE.DBF
53 ERASE TEMPONE.NDX
54 SET CONSOLE ON
55 *
56 SET COLOR TO W+/BR, W+/BR
57 @ 13,15 SAY "Enter site number for which the report is desired:"
58 USE MANUAL INDEX MANULSIT
59 *
60 DO WHILE .T.
61     SET COLOR TO /BR, /BR
62     STORE LOSITE TO MSITE
63     @ 13,66 GET MSITE PICT '99'
64     READ
65     IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
66         SET COLOR TO W+/R, W+/R
67         STORE ' Response must be between ' + LOSITE + ;
68             ' and ' + HISITE + ' ' TO ERROR
69         @ 24,22 SAY ERROR
70         DO DELAY
71         LOOP
72     ELSE
73         GO TOP
74         FIND &MSITE
75         IF EOF() = .T. THEN
76             STORE " No manuals exist for site " + MSITE + ;
77                 ", try another site " TO MESSAGE
78             SET COLOR TO W+/R, W+/R
79             @ 24,16 SAY MESSAGE
80             DO DELAY
81             LOOP
82         ELSE
83             EXIT
84         ENDIF EOF() = .T.
85     ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
86 ENDDO WHILE .T.
87 *
88 SET COLOR TO /BR, /BR
89 @ 13,15 SAY SPACE(55)
90 *
91 SET COLOR TO R+/ , R+/
92 @ 13,13 SAY " CREATING THE TEMPORARY DATABASE AND ASSOCIATED INDEX "
93 *
94 * CREATE THE TEMPORARY DATABASE TO BE USED
95 *
96 SET CONSOLE OFF
97 COPY STRUCTURE TO TEMPONE
98 USE TEMPONE
99 APPEND FROM MANUAL FOR SITENO = "&MSITE"
100 INDEX ON FEATURENO TO TEMPONE
```

```
101 SET CONSOLE ON
102 *
103 * CREATE THE SPLICE EQUIPMENT PROJECT REPORT AND CHECK IF THE REPORT
104 * IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
105 *
106 SET COLOR TO /BR, /BR
107 @ 13,12 SAY SPACE(65)
108 *
109 SET COLOR TO W+/BR, W+/BR
110 @ 13,16 SAY " Do you want a printed report? (Yes or No): "
111 SET COLOR TO /BR, /BR
112 @ 13,49 SAY "Y"
113 @ 13,56 SAY "N"
114 STORE "N" TO ACCEPT
115 @ 13,62 GET ACCEPT PICT "!"
116 READ
117 *
118 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
119 *
120 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
121 IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
122 SET COLOR TO W+/R,W+/R
123 @ 24,24 SAY " Response must be either N or Y "
124 DO DELAY
125 STORE "N" TO ACCEPT
126 ENDIF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
127 SET COLOR TO /BR, /BR
128 @ 13,62 GET ACCEPT PICT "!"
129 READ
130 ENDDO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
131 *
132 SET COLOR TO /BR, /BR
133 @ 13,12 SAY SPACE(65)
134 *
135 SELECT 1
136 USE TEMPONE
137 SELECT 2
138 USE DESCRIP INDEX DESCRIP
139 SELECT TEMPONE
140 SET RELATION TO FEATURENO INTO DESCRIP
141 GO TOP
142 *
143 @ 13,12 SAY SPACE(65)
144 *
145 IF ACCEPT = "Y" THEN
146 ?? FLASH + "W.PRINTER/"
147 SET CONSOLE OFF
148 WAIT TO CHOICE
149 SET CONSOLE ON
150 SET COLOR TO W/B, W/B
```



```
201 SET COLOR TO GR+/B, GR+/B
202 @ 5,2 SAY "SITE CLIN FEATURE# DESCRIPTION"
203 @ 5,57 SAY "MANUAL DESCRIPTION"
204 SET COLOR TO /BR, /BR
205 STORE 0 TO LINECT
206 *
207 DO WHILE .NOT. EOF()
208     DO WHILE LINECT < 15
209         @ LINECT+7,3 SAY SITENO
210         @ LINECT+7,8 SAY DESCRIP->CLIN
211         @ LINECT+7,16 SAY FEATURENO
212         @ LINECT+7,25 SAY DESCRIP->DESCRIPT
213         @ LINECT+7,54 SAY MANLDESC
214         LINECT = LINECT + 1
215         SKIP
216         IF EOF() = .T. THEN
217             SET COLOR TO W+/R, W+/R
218             @ 24,18 SAY " End of File reached, Press any key to EXIT "
219             SET CONSOLE OFF
220             WAIT TO ACCEPT
221             SET CONSOLE ON
222             EXIT
223         ENDIF EOF() = .T.
224     ENDDO WHILE LINECT < 15
225 *
226 IF EOF() = .T. THEN
227     EXIT
228 ENDIF EOF() = .T.
229 SET COLOR TO R+/B, R+/B
230 STORE "C" TO CHOICE
231 @ 22,57 GET CHOICE PICT "!"
232 READ
233 *
234 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
235 *
236 DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
237     IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
238         SET COLOR TO W+/R, W+/R
239         @ 24,24 SAY " Response must be either C or X "
240         DO DELAY
241         STORE "C" TO CHOICE
242     ENDIF .NOT. (CHOICE = "C" .OR. CHOICE = "X")
243     SET COLOR TO R+/B, R+/B
244     @ 22,57 GET CHOICE PICT "!"
245     READ
246 ENDDO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
247 *
248 * DETERMINE IF THE USER WANTS TO QUIT OR CONTINUE
249 *
250 IF CHOICE = "C"
```

Page 6

MNLSTRPT.PRG Program Listing

```
251      SET COLOR TO /BR, /BR
252      @ 07,2 SAY SPACE(76)
253      @ 08,2 SAY SPACE(76)
254      @ 09,2 SAY SPACE(76)
255      @ 10,2 SAY SPACE(76)
256      @ 11,2 SAY SPACE(76)
257      @ 12,2 SAY SPACE(76)
258      @ 13,2 SAY SPACE(76)
259      @ 14,2 SAY SPACE(76)
260      @ 15,2 SAY SPACE(76)
261      @ 16,2 SAY SPACE(76)
262      @ 17,2 SAY SPACE(76)
263      @ 18,2 SAY SPACE(76)
264      @ 19,2 SAY SPACE(76)
265      @ 20,2 SAY SPACE(76)
266      @ 21,2 SAY SPACE(76)
267      STORE 0 TO LINECT
268      ELSE
269      EXIT
270      ENDIF CHOICE = "C"
271      *
272      ENDDO WHILE .NOT. EOF()
273      *
274      ENDIF ACCEPT = "Y"
275      *
276      * ERASE THE TEMPORARY DATABASE AND ASSOCIATED INDEX USED FOR TOTALS
277      *
278      CLOSE DATABASES
279      SET CONSOLE OFF
280      ERASE TEMPONE.DBF
281      ERASE TEMPONE.NDX
282      SET CONSOLE ON
283      SET PRINT OFF
284      *
285      * RETURN TO CALLING PROGRAM
286      *
287      RELEASE ACCEPT, CHOICE, ERROR, LINECT, MSITE, PAGENO, TODAY, TODATE
288      RETURN
289      *****
```

Page 1

NEWDOADD.PRG Program Listing

```

1  * PROCEDURE NEWDOADD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              LCDR WINSTON H. BUCKLEY, SC, USN
5  *              LCDR ROBERT F. BRADO, USN
6  *              LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : TO ADD A NEW DELIVERY ORDER TO THE EXISTING EQUIPMENT,
9  *              MANUAL, AND SERIAL NUMBER DATA BASES WHILE UPDATING
10 *              ALL INDEXES. NO TEMP.DBF LINE ITEM WITH A BLANK
11 *              OR "XXXXXX" FEATURE NUMBER WILL BE ADDED TO THE FILE.
12 *
13 * INPUT FILES   : EQUIP.DBF, MANUAL.DBF, SERIALNO.DBF, TEMP.DBF,
14 *              NEWDOMOD.DBF, SERNOTMP.DBF.
15 *
16 * OUTPUT FILES  : EQUIP.DBF, MANUAL.DBF, SERIALNO.DBF, TEMP.DBF,
17 *              NEWDOMOD.DBF, SERNOTMP.DBF.
18 *
19 * CALLED BY     : NEWDOCVT.PRG
20 *
21 * MODULES CALLED : SERNOBLD.PRG
22 *
23 * LOCAL VARIABLES: MESSAGE, MFEAT, MSITE, MINDEX
24 *
25 * DATE LAST TIME MODIFIED =====> 22 DECEMBER 1985 <=====
26 *
27 * CLEAR SCREEN. COPY AND MODIFY INPUT FILE TO DATABASE FORMAT FOR
28 * ALL RECORDS THAT DON'T HAVE A BLANK OR "XXXXXX" IN THE FEATURE NUMBER.
29 *
30 SET COLOR TO R+ / , R+ /
31 @ 15,24 SAY " UPDATING THE EQUIPMENT DATABASE "
32 STORE "      Adding new records to the EQUIPMENT database" +;
33 " , PLEASE BE PATIENT      " TO MESSAGE
34 SET COLOR TO /W, /W
35 @ 24,0 SAY MESSAGE
36 USE EQUIP
37 COPY STRUCTURE TO NEWDOMOD
38 USE NEWDOMOD
39 APPEND FROM TEMP.DBF FOR FEATURENO <> ' ' .AND. FEATURENO <> 'XXXXXX'
40 *
41 * FILL-IN THE EFFECTIVE DELIVERY ORDER DATE FIELD WITH THE DATE SUPPLIED
42 * BY THE USER AND ADD THE DELIVERY ORDER TO THE EQUIPMENT DATABASE.
43 *
44 REPLACE ALL EFFDATE WITH MEFFDATE
45 USE EQUIP INDEX EQUIPDAT, EQUIPSIT, EQUIPPRJ, EQUIPSD, EFEAT
46 APPEND FROM NEWDOMOD
47 *
48 * ADDING THE NEW RECORDS FOR THE MANUAL DATABASE.
49 *
50 SET COLOR TO R+ / , R+ /

```

Page 2

NEWDOADD.PRG Program Listing

```

51 @ 15,24 SAY "  UPDATING THE MANUALS DATABASE  "
52 STORE "      Adding new records to the MANUAL database" +;
53 " , PLEASE BE PATIENT      " TO MESSAGE
54 SET COLOR TO /w, /w
55 @ 24,0 SAY MESSAGE
56 CLOSE DATABASES
57 SELECT 1
58 USE MANUAL INDEX MANULSIT
59 SELECT 2
60 USE NEWDOMOD
61 *
62 DO WHILE .NOT. EOF()
63     STORE SITENO TO MSITE
64     STORE FEATURENO TO MFEAT
65     STORE SITENO + FEATURENO TO MINDEX
66     SELECT 1
67     GO TOP
68     FIND &MINDEX
69     IF EOF( )
70         GO BOTTOM
71         INSERT BLANK
72         REPLACE FEATURENO WITH "&MFEAT"
73         REPLACE SITENO WITH "&MSITE"
74     ENDIF
75     SELECT 2
76     SKIP
77 ENDDO WHILE .NOT. EOF()
78 *
79 * BUILDING A DUMMY SERIAL NUMBER FILE WHICH WILL BE MODIFIED AND
80 * EXPANDED WHEN ALL DELIVERY ORDERS HAVE BEEN LOADED.
81 *
82 SET COLOR TO R+ / , R+ /
83 @ 15,24 SAY " BUILDING THE SERIAL NUMBER FILE  "
84 STORE "      Adding new records to the SERIAL NUMBER database, " +;
85 " PLEASE BE PATIENT      " TO MESSAGE
86 SET COLOR TO /w, /w
87 @ 24,0 SAY MESSAGE
88 USE SERIALNO
89 COPY STRUCTURE TO SERNOIMP
90 USE SERNOIMP
91 APPEND FROM NEWDOMOD
92 *
93 * CALL THE PROGRAM TO BUILD THE BLANK SERIAL NUMBER RECORDS
94 *
95 DO SERNOBLD
96 STORE " Appending new records to the database may be a long process," +;
97 " PLEASE BE PATIENT      " TO MESSAGE
98 SET COLOR TO /w, /w
99 @ 24,0 SAY MESSAGE
100 SET COLOR TO R+ / , R+ /

```

```

101 @ 15,12 SAY " APPENDING NEW RECORDS TO THE SERIAL NUMBER DATABASE "
102 USE SERIALNO INDEX SERNOPRJ, SERNOSIT, SERNODAT, SERNOFEA
103 APPEND FROM SERNOTMP
104 SET COLOR TO W/B, W/B
105 @ 15,10 SAY SPACE(65)
106 @ 24,0 SAY SPACE(80)
107 *
108 * RETURNING TO THE CALLING PROGRAM.
109 *
110 CLOSE DATABASES
111 RETURN
112 *****

```

```
1 * PROCEDURE NEWDOCMD.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *               LCDR WINSTON H. BUCKLEY, SC, USN
5 *               LCDR ROBERT F. BRADO, USN
6 *               LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : LOAD NEW DELIVERY ORDERS TO THE DATABASE FILES.
9 *
10 * INPUT FILES  : NONE.
11 *
12 * OUTPUT FILES : NONE.
13 *
14 * CALLED BY    : MAINMENU.PRG.
15 *
16 * MODULES CALLED : NEWDOCVT.PRG
17 *
18 * LOCAL VARIABLES: SELEKT
19 *
20 * DATE LAST TIME MODIFIED =====> 22 DECEMBER 1985 <=====
21 *
22 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR SELECTION
23 *
24 STORE "1" TO SELEKT
25 DO WHILE SELEKT < "2"
26     SET COLOR TO W/B, W/B
27     CLEAR
28     ?? FLASH + "W.NEWDOCMD/"
29     SET CONSOLE OFF
30     WAIT TO SELEKT
31     SET CONSOLE ON
32 *
33 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
34 *
35 DO CASE
36 *
37 *     CALL THE NEW DELIVERY ORDER CONVERT AND LOAD PROGRAM.
38 *     CASE SELEKT = "1"
39 *         DO NEWDOCVT
40 *
41 *     RETURN TO THE MAINMENU PROGRAM.
42 *     CASE SELEKT = "2"
43 *
44 ENDCASE
45 *
46 ENDDO (WHILE SELEKT = "2")
47 *
48 * RETURN TO THE CALLING PROGRAM
49 *
50 RETURN
```

Page 2

NEWDOCMD.PRG Program Listing

51 | *****

Page 1

NEWDOCVT.PRG Program Listing

```
1 * PROCEDURE NEWDOCVT.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. BUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : TO COMPARE AN INCOMING NEW DELIVERY ORDER TO THE
9 *              : EXISTING EQUIPMENT DATABASE AND CHECK FOR DUPLICATE
10 *             : SITE NUMBER AND DELIVERY ORDER DATE. IF THE SITE
11 *             : NUMBER AND DELIVERY DATE ARE UNIQUE OR THE USER
12 *             : DECIDES TO LOAD THE DUPLICATED SITE NUMBER/DELIVERY
13 *             : ORDER ANYWAY THEN THE NEW DELIVERY ORDER ADD
14 *             : PROGRAM IS CALLED. WHEN ALL DELIVERY ORDERS ARE
15 *             : ADDED THEN SPECIFIED INDEXES ARE UPDATED.
16 *
17 * INPUT FILES  : TED.DBF, EQUIP.DBF, MANUAL.DBF, SERIALNO.DBF, NEW
18 *              : DELIVERY ORDER .PRN FILE, EFFDATE.NDX, EQUIPSIT.NDX,
19 *              : EQUIPPRJ.NDX, MANULSIT.NDX, SERNOPRJ.NDX, SERNOSIT.NDX,
20 *              : SERNODAT.NDX, NEWDOMOD.DBF, TEMP.DBF, SERNOIMP.DBF,
21 *              : EFEBAT.NDX
22 *
23 * OUTPUT FILES : EQUIP.DBF, MANUAL.DBF, SERIAL.DBF, EFFDATE.NDX,
24 *              : EQUIPSIT.NDX, EQUIPPRJ.NDX, MANULSIT.NDX, EFEBAT.NDX
25 *              : SERNOPRJ.NDX, SERNOSIT.NDX, SERNODAT.NDX.
26 *
27 * CALLED BY    : NEWDOCMD.PRG
28 *
29 * MODULES CALLED : NEWDOADD.PRG, DELAY.PRG
30 *
31 * GLOBAL VARIABLE: HDATE, HSITE, LDATE, LOSITE
32 *
33 * LOCAL VARIABLES: ACCEPT, CHOICE, DBNAME, ERASIT, ERROR, MDAY, MEFFDATE,
34 *              : MESSAGE, MKEY, MMONTH, MOLDATE, MSITE, MYEAR, NOFILE
35 *
36 * DATE LAST TIME MODIFIED =====> 22 DECEMBER 1985 <=====
37 *
38 * SET UP INITIAL STRUCTURE AND RECEIVE INPUT INFORMATION.
39 * AND START LOOP PROCESS.
40 *
41 SET ESCAPE OFF
42 SET TALK OFF
43 SET COLOR TO W+/B, W+/B, B
44 ?? FLASH +"S.NEWDOCVT.SCR/"
45 @ 24,0 SAY SPACE(80)
46 STORE "Are all input entries correct? (Yes or No):" to CORRECT
47 DO WHILE .T.
48 *
49 *   OBTAIN THE INPUT VALUES FROM THE USER
50 *
```



```

51 DO WHILE .T.
52     STORE SPACE(18) + "Enter the name of the .PRN file to be loaded" +;
53     SPACE(18) TO MESSAGE
54     SET COLOR TO /W, /W
55     @ 24,0 SAY MESSAGE
56     STORE "SPLICE " TO DBNAME
57     STORE DTOC(DATE()) TO SYSDATE
58     STORE SUBSTR(SYSDATE,7,2) + SUBSTR(SYSDATE,1,2) +;
59     SUBSTR(SYSDATE,4,2) TO MEFFDATE
60     STORE "01" TO MSITE
61     SET COLOR TO /BR, /BR
62     @ 6,54 GET DBNAME PICT "!!!!!!!"
63     READ
64     STORE 0 TO NOFILE
65 *
66 DO WHILE .NOT. FILE(TRIM(DBNAME)+".PRN")
67     SET COLOR TO W/B, W/B
68     @ 24,0 SAY SPACE(80)
69     SET COLOR TO W+/R, W+/R
70     @ 24,24 SAY " File does not exist, try again "
71     DO DELAY
72     NOFILE = NOFILE + 1
73     IF NOFILE = 3 THEN
74         SET COLOR TO W+/BG, W+/BG
75         @ 17,15 SAY " Do you want to exit this process? (Yes or No): "
76         SET COLOR TO /BG, /BG
77         @ 17,51 SAY "Y"
78         @ 17,58 SAY "N"
79         STORE "Y" TO ACCEPT
80         @ 17,63 GET ACCEPT PICT "!"
81         READ
82         DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
83             IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
84                 SET COLOR TO W+/R, W+/R
85                 @ 24,24 SAY " Response must be either N or Y "
86                 DO DELAY
87                 STORE "Y" TO ACCEPT
88             ENDIF
89             SET COLOR TO /BG, /BG
90             @ 17,63 GET ACCEPT PICT "!"
91             READ
92         ENDDO
93         IF ACCEPT = "Y" THEN
94             SET CONSOLE OFF
95             CLOSE DATABASES
96             ERASE TEMP.DBF
97             ERASE NEWLOMOD.DBF
98             ERASE SERNOTMP.DBF
99             SET CONSOLE ON
100            RELEASE ALL LIKE M*, ACCEPT, CHOICE, CORRECT, DBNAME,;

```

```
101          ERASIT, NOFILE, SYSDATE
102          RETURN
103          ELSE
104              NOFILE = 0
105          ENDIF
106          SET COLOR TO W+/B, W+/B
107          @ 17,10 SAY SPACE(55)
108      ENDIF
109      SET COLOR TO /W, /W
110      @ 24,0 SAY MESSAGE
111      STORE "SPLICE " TO DBNAME
112      SET COLOR TO /BR, /BR
113      @ 6,54 GET DBNAME PICT "!!!!!!!"
114      READ
115  ENDDO
116  STORE TRIM(DBNAME) + ".PRN" TO DBNAME
117  USE TED
118  COPY TO TEMP.DBF
119  USE TEMP.DBF
120  APPEND FROM &DBNAME SDF
121  GO TOP
122  *
123  *   HAVE THE USER SPECIFY THE EFFECTIVE DATE OF THE DELIVERY ORDER
124  *
125  STORE SPACE(17) + "Input Effective Date (Range " + LODATE + ;
126  " to " + HIDATE + ")" + SPACE(17) TO MESSAGE
127  SET COLOR TO /W, /W
128  @ 24,0 SAY MESSAGE
129  STORE "000000" TO MOLDATE
130  DO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDATE)
131      STORE MEFFDATE TO MOLDATE
132      SET COLOR TO /BR, /BR
133      @ 8,54 GET MOLDATE PICT "999999"
134      READ
135      DO WHILE .T.
136          IF .NOT. (SUBSTR(MOLDATE,1,2) > "83" .AND. ;
137              SUBSTR(MOLDATE,1,2) <= "99") THEN
138              SET COLOR TO W/B, W/B
139              @ 24,0 SAY SPACE(80)
140              SET COLOR TO W+/R, W+/R
141              @ 24,16 SAY " Year portion of date must be between 84 and 99 "
142              DO DELAY
143              SET COLOR TO /W, /W
144              @ 24,0 SAY MESSAGE
145              STORE SUBSTR(MEFFDATE,1,2) TO MYEAR
146              SET COLOR TO /BR, /BR
147              @ 8,54 GET MYEAR PICT "99"
148              READ
149              STORE MYEAR + SUBSTR(MOLDATE,3,4) TO MOLDATE
150          LOOP
```

```

151      ELSE
152          EXIT
153      ENDIF
154      ENDDO WHILE .T.
155  *
156      DO WHILE .T.
157      IF .NOT. (SUBSTR(MOLDATE,3,2) >= "01" .AND.;
158              SUBSTR(MOLDATE,3,2) <= "12") THEN
159          SET COLOR TO W/B, W/B
160          @ 24,0 SAY SPACE(80)
161          SET COLOR TO W+/R, W+/R
162          @ 24,16 SAY " Month portion of date must be between 01 and 12 "
163          DO DELAY
164          SET COLOR TO /W, /W
165          @ 24,0 SAY MESSAGE
166          STORE SUBSTR(MEFFDATE,3,2) TO MMONTH
167          SET COLOR TO /BR, /BR
168          @ 8,56 GET MMONTH PICT "99"
169          READ
170          STORE SUBSTR(MOLDATE,1,2) + MMONTH +;
171              SUBSTR(MOLDATE,5,2) TO MOLDATE
172          LOOP
173      ELSE
174          EXIT
175      ENDIF
176      ENDDO WHILE .T.
177  *
178      DO WHILE .T.
179      IF ((SUBSTR(MOLDATE,3,2) = "04" .OR. SUBSTR(MOLDATE,3,2) = "06" .OR.;
180          SUBSTR(MOLDATE,3,2) = "09" .OR. SUBSTR(MOLDATE,3,2) = "11") .AND.;
181      .NOT.(SUBSTR(MOLDATE,5,2) >= "01" .AND. SUBSTR(MOLDATE,5,2) <= "30"))
182          SET COLOR TO W/B, W/B
183          @ 24,0 SAY SPACE(80)
184          SET COLOR TO W+/R, W+/R
185          @ 24,16 SAY " Day portion of date must be between 01 and 30 "
186          DO DELAY
187          SET COLOR TO /W, /W
188          @ 24,0 SAY MESSAGE
189          STORE SUBSTR(MEFFDATE,5,2) TO MDAY
190          SET COLOR TO /BR, /BR
191          @ 8,58 GET MDAY PICT "99"
192          READ
193          STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
194          LOOP
195      ELSE
196  *
197      IF (SUBSTR(MOLDATE,3,2) = "02" .AND. .NOT.;
198          (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
199          SUBSTR(MOLDATE,5,2) <= "28")) THEN
200          SET COLOR TO W/B, W/B

```

```

201      @ 24,0 SAY SPACE(80)
202      SET COLOR TO W+/R, W+/R
203      @ 24,16 SAY " Day portion of date must be between 01 and 28 "
204      DO DELAY
205      SET COLOR TO /W, /W
206      @ 24,0 SAY MESSAGE
207      STORE SUBSTR(MEFFDATE,5,2) TO MDAY
208      SET COLOR TO /BR, /BR
209      @ 8,58 GET MDAY PICT "99"
210      READ
211      STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
212      LOOP
213      ELSE
214      *
215      IF .NOT. (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
216          SUBSTR(MOLDATE,5,2) <= "31") THEN
217          SET COLOR TO W/B, W/B
218          @ 24,0 SAY SPACE(80)
219          SET COLOR TO W+/R, W+/R
220          @ 24,16 SAY " Day portion of date must be between 01 and 31 "
221          DO DELAY
222          SET COLOR TO /W, /W
223          @ 24,0 SAY MESSAGE
224          STORE SUBSTR(MEFFDATE,5,2) TO MDAY
225          SET COLOR TO /BR, /BR
226          @ 8,58 GET MDAY PICT "99"
227          READ
228          STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
229          LOOP
230      ELSE
231      EXIT
232      ENDIF
233      ENDIF
234      ENDIF
235      ENDDO WHILE .T.
236      ENDDO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDEATE)
237      *
238      STORE MOLDATE TO MEFFDATE
239      SET COLOR TO W/B, W/B
240      @ 24,0 SAY SPACE(80)
241      STORE SPACE(8) + "Enter site number of Delivery Order to be " +;
242          "loaded to the database" + SPACE(8) TO MESSAGE
243      SET COLOR TO /W, /W
244      @ 24,0 SAY MESSAGE
245      SET COLOR TO /BR, /BR
246      @ 11,54 SAY SITENO PICT "99"
247      @ 13,54 GET MSITE PICT "99"
248      READ
249      *
250      DO WHILE .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)

```

```

251      IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
252          SET COLOR TO W/B, W/B
253          @ 24,0 SAY SPACE(80)
254          SET COLOR TO W+/R, W+/R
255          STORE ' Response must be between ' + LOSITE +;
256              ' and ' + HISITE + ' ' TO ERROR
257          @ 24,21 SAY ERROR
258          DO DELAY
259          SET COLOR TO /W, /W
260          @ 24,0 SAY MESSAGE
261          SET COLOR TO /BR, /BR
262          STORE '01' TO MSITE
263          @ 13,54 GET MSITE PICT "99"
264          READ
265      ENDIF
266  ENDDO
267  *
268  *      ASK THE USER IF THE INPUTS ARE VALID OR NOT
269  *
270      SET COLOR TO W+/B, W+/B
271      @ 24,0 SAY SPACE(80)
272      @ 16,17 SAY CORRECT
273      SET COLOR TO R+/B, R+/B
274      @ 16,49 SAY "Y"
275      @ 16,56 SAY "N"
276      STORE "N" TO ACCEPT
277      @ 16,62 GET ACCEPT PICT "!"
278      READ
279  *
280  *      ENSURE THAT THE RESPONSE IS EITHER "Y" OR "N"
281  *
282      DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
283          IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
284              SET COLOR TO W+/R, W+/R
285              @ 24,24 SAY " Response must be either N or Y "
286              DO DELAY
287              STORE "N" TO ACCEPT
288          ENDIF
289          SET COLOR TO R+/B, R+/B
290          @ 16,62 GET ACCEPT PICT "!"
291          READ
292      ENDDO
293      SET COLOR TO W+/B, W+/B
294      @ 16,15 SAY SPACE(55)
295  *
296      IF ACCEPT = "Y" THEN
297  *
298  *          ASK THE USER IF THE INPUT ".PRN" FILE IS TO BE ERASED
299  *
300      SET COLOR TO W+/B, W+/B

```

```

301      STORE "Do you want to erase the input file " + DBNAME +;
302      " ? (Yes or No): " TO MESSAGE
303      @ 16,10 SAY MESSAGE
304      STORE "N" TO ERASIT
305      SET COLOR TO R+/B, R+/B
306      @ 16,46 SAY DBNAME
307      @ 16,45+LEN(DBNAME)+5 SAY "Y"
308      @ 16,45+LEN(DBNAME)+12 SAY "N"
309      @ 16,45+LEN(DBNAME)+17 GET ERASIT PICT "!"
310      READ
311      *
312      *      ENSURE THAT THE RESPONSE IS EITHER "Y" OR "N"
313      *
314      DO WHILE .NOT. (ERASIT = "N" .OR. ERASIT = "Y")
315          IF .NOT. (ERASIT = "N" .OR. ERASIT = "Y") THEN
316              SET COLOR TO W+/R, W+/R
317              @ 24,24 SAY " Response must be either N or Y "
318              DO DELAY
319              STORE "N" TO ERASIT
320          ENDIF
321          SET COLOR TO R+/B, R+/B
322          @ 16,45+LEN(DBNAME)+17 GET ERASIT PICT "!"
323          READ
324      ENDDO
325      SET COLOR TO W+/B, W+/B
326      @ 16,10 SAY SPACE(65)
327      *
328      IF ERASIT = "Y" THEN
329          ERASE &DBNAME
330      ENDIF
331      EXIT
332      ELSE
333          SET COLOR TO /BR, /BR
334          @ 8,54 SAY " "
335          @ 11,54 SAY " "
336          @ 13,54 SAY " "
337          LOOP
338      ENDIF
339      ENDDO WHILE .T.
340      *
341      SET COLOR TO W+/B, W+/B
342      @ 16,10 SAY SPACE(65)
343      REPLACE ALL SITE#O WITH "&MSITE"
344      USE EQUIP INDEX EQUIPSD
345      STORE MEFFDATE + MSITE TO MKEY
346      GO TOP
347      IF EOF() = .T. THEN
348          DO NEWDOADD
349      ELSE
350          FIND &MKEY

```

```
351 IF EOF() = .T. THEN
352 DO NEWDOADD
353 ELSE
354 SET COLOR TO R+*/ , R+*/
355 @ 16,21 SAY " THIS IS A DUPLICATE DELIVERY ORDER! "
356 SET COLOR TO W+/B, W+/B
357 @ 17,17 SAY " Do you still desire to load it? (Yes or No): "
358 SET COLOR TO R+/B, R+/B
359 @ 17,51 SAY "Y"
360 @ 17,58 SAY "N"
361 STORE "N" TO ACCEPT
362 @ 17,63 GET ACCEPT PICT "!"
363 READ
364 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
365 IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
366 SET COLOR TO W+/R, W+/R
367 @ 24,24 SAY " Response must be either N or Y "
368 DO DELAY
369 STORE "N" TO ACCEPT
370 ENDIF
371 SET COLOR TO R+/B, R+/B
372 @ 17,63 GET ACCEPT PICT "!"
373 READ
374 ENDDO
375 SET COLOR TO W/B, W/B
376 @ 16,20 SAY SPACE(50)
377 @ 17,15 SAY SPACE(55)
378 IF ACCEPT = "Y" THEN
379 DO NEWDOADD
380 ENDIF
381 ENDIF
382 ENDIF
383 *
384 * CHECK TO SEE IF THERE ARE MORE DELIVERY ORDERS TO BE ADDED.
385 *
386 SET COLOR TO R+/B, R+/B
387 STORE "N" TO CHOICE
388 @ 21,68 GET CHOICE PICT "!"
389 READ
390 *
391 * ENSURE THAT THE RESPONSE IS EITHER "Y" OR "N"
392 *
393 DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "Y")
394 IF .NOT. (CHOICE = "N" .OR. CHOICE = "Y") THEN
395 SET COLOR TO W+/R, W+/R
396 @ 24,24 SAY " Response must be either N or Y "
397 DO DELAY
398 STORE "N" TO CHOICE
399 ENDIF
400 SET COLOR TO R+/B, R+/B
```

```

401      @ 21,68 GET CHOICE PICT "!"
402      READ
403      ENDDO
404      *
405      IF CHOICE = "N" THEN
406          EXIT
407      ELSE
408          SET COLOR TO W/B, W/B
409          @ 19,10 SAY SPACE(65)
410          @ 21,68 SAY " "
411          SET COLOR TO /BR, /BR
412          @ 8,54 SAY " "
413          @ 11,54 SAY " "
414          @ 13,54 SAY " "
415      ENDIF
416      *
417      ENDDO WHILE .T.
418      *
419      * ERASE ALL TEMPORARY DBF FILES CREATED DURING THE LOAD
420      *
421      SET COLOR TO R+/ , R+/
422      @ 15,26 SAY " ERASING TEMPORARY DATABASES "
423      CLOSE DATABASES
424      SET CONSOLE OFF
425      ERASE TEMP.DBF
426      ERASE NEWDOMOD.DBF
427      ERASE SERNOTMP.DBF
428      SET CONSOLE ON
429      *
430      * RETURN TO THE CALLING PROGRAM
431      *
432      RELEASE ALL LIKE M*, ACCEPT, CHOICE, CORRECT, DBNAME, ERASIT, ERROR,;
433      NOFILE, SYSDATE
434      RETURN
435      *****

```


Page 1

PROJRPTS.PRG Program Listing

```

1  * PROCEDURE PROJRPTS.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              LCDR WINSTON H. BUCKLEY, SC, USN
5  *              LCDR ROBERT F. BRADO, USN
6  *              LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER A SELECTION OF PROJECT LEVEL REPORTS.
9  *
10 * INPUT FILES   : NONE.
11 *
12 * OUTPUT FILES  : NONE.
13 *
14 * CALLED BY     : REPORCMD.PRG
15 *
16 * MODULES CALLED : EQPPJRPT.PRG, SNOJRPT.PRG
17 *
18 * LOCAL VARIABLES: PROJRPTS
19 *
20 * DATE LAST TIME MODIFIED =====> 18 DECEMBER 1985 <=====
21 *
22 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE SELECTION.
23 *
24 STORE "1" TO PROJRPTS
25 DO WHILE PROJRPTS < "3"
26     SET COLOR TO W/B, W/B, B
27     CLEAR
28     ?? FLASH + "W.PROJRPTS/"
29     SET CONSOLE OFF
30     WAIT TO PROJRPTS
31     SET CONSOLE ON
32 *
33 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
34 *
35     DO CASE
36 *
37 *         CALL THE EQUIPMENT PROJECT LEVEL REPORT.
38         CASE PROJRPTS = "1"
39             DO EQPPJRPT
40 *
41 *         CALL THE SERIAL NUMBER PROJECT LEVEL REPORT.
42         CASE PROJRPTS = "2"
43             DO SNOJRPT
44 *
45 *         RETURN TO THE SPLICE REPORTING LEVEL MENU.
46         CASE PROJRPTS = "3"
47 *
48     ENDCASE
49 *
50 ENDDO (WHILE PROJRPTS < "3")

```

Page 2

PROJRPTS.PRG Program Listing

```
51 | *  
52 | * RETURN TO THE CALLING PROGRAM  
53 | *  
54 | RETURN  
55 | *****
```

```

1  * PROCEDURE REPORCMD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER AN OPPORTUNITY TO SELECT A REPORT
9  *               LEVEL - PROJECT LEVEL, SITE LEVEL, OR DELIVERY ORDER
10 *               DATE LEVEL.
11 *
12 * INPUT FILES   : NONE.
13 *
14 * OUTPUT FILES  : NONE.
15 *
16 * MODULES CALLED : PROJRPPTS.PRG, SITRPPTS.PRG, DATRPPTS.PRG
17 *
18 * LOCAL VARIABLES: SELEKT
19 *
20 * DATE LAST TIME MODIFIED =====> 22 DECEMBER 1985 <=====
21 *
22 * DISPLAY THE REPORT LEVEL MENU TO THE USER AND WAIT FOR THE SELECTION.
23 *
24 STORE "1" TO SELEKT
25 DO WHILE SELEKT < "4"
26     SET COLOR TO W/B, W/B, B
27     CLEAR
28     ?? FLASH + "W.REPORCMD/"
29     SET CONSOLE OFF
30     WAIT TO SELEKT
31     SET CONSOLE ON
32 *
33 * PROTECT ROUTINE BASED ON THE USER'S SELECTION.
34 *
35 DO CASE
36 *
37 *     CALL THE PROJECT LEVEL REPORTS PROGRAM.
38     CASE SELEKT = "1"
39         DO PROJRPPTS
40 *
41 *     CALL THE SITE LEVEL REPORTS PROGRAM.
42     CASE SELEKT = "2"
43         DO SITRPPTS
44 *
45 *     CALL THE EFFECTIVE DELIVERY ORDER DATE LEVEL REPORTS PROGRAM.
46     CASE SELEKT = "3"
47         DO DATRPPTS
48 *
49 *     RETURN TO THE MAIN MENU PROGRAM.
50     CASE SELEKT = "4"

```

Page 2

REPORCMD.PRG Program Listing

```
51 | *
52 |   ENDCASE
53 | *
54 | ENDDO (WHILE SELEKT < "4")
55 | *
56 | *   RETURN TO THE CALLING PROGRAM
57 | *
58 | RETURN
59 | *****
```

Page 1

SELECTOR.PRG Program Listing

```

1  * PROCEDURE NAME : SELECTOR.PRG
2  *
3  * AUTHORS       : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE       : TO PERMIT THE USER TO SELECT THE DESIRED PROCESSING
9  *               ACTION. CHOICES INCLUDE: THE SPLICE CONFIGURER,
10 *               LOTUS 1-2-3 FOR "WHAT-IF" ANALYSIS, AND THE dBASE III
11 *               SPLICE CONFIGURATION MANAGEMENT SYSTEM. CHANGES TO
12 *               ACTIVE DIRECTORIES AND CALLS TO dBASE EXTERNAL PROGRAMS
13 *               ARE EFFECTED WITH THE dBASE "RUN" COMMAND.
14 *
15 * INPUT FILES   : NONE.
16 *
17 * OUTPUT FILES  : NONE.
18 *
19 * MODULES CALLED : SPLICE.COM; 123.EXE, MAINMENU.PRG, DELAY.PRG, WS.COM
20 *
21 * GLOBAL VARIABLE: FLASH
22 *
23 * LOCAL VARIABLES: ANS
24 *
25 * DATE LAST TIME MODIFIED =====> 18 DECEMBER 1985 <=====
26 *
27 * DBASE PROGRAM CONFIGURATION VARIABLES:
28 *
29 SET BELL OFF
30 SET CONSOLE ON
31 SET INTENSITY OFF
32 SET SCOREBOARD OFF
33 SET TALK OFF
34 *
35 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE USER'S CHOICE.
36 *
37 PUBLIC FLASH
38 ?? CHR(145) + "L.SPLICE.WIN/"
39 STORE "1" TO ANS
40 DO WHILE .T.
41     SET COLOR TO W+/B, W+/B, B
42     CLEAR
43     FLASH = CHR(145)
44     ?? FLASH + "S.SELECTOR.SCR/"
45     @ 24,0 SAY SPACE (80)
46     SET COLOR TO R+/B,R+/B
47     @ 21,53 GET ANS PICT "9"
48     READ
49     DO WHILE (ANS < "1" .OR. ANS > "6")
50         IF (ANS < "1" .OR. ANS > "6") THEN

```

Page 2

SELECTOR.PRG Program Listing

```
51      SET COLOR TO W+/R,W+/R
52      @ 24,23 SAY " Response must be between 1 and 6 "
53      DO DELAY
54      STORE "1" TO ANS
55      ENDIF
56      SET COLOR TO R+/B,R+/B
57      @ 21,53 GET ANS PICT "9"
58      READ
59      ENDDO
60      *
61      * PERFORM APPROPRIATE TASK BASED ON THE USER'S CHOICE.
62      *
63      DO CASE
64      *
65      * CHANGE THE ACTIVE DIRECTORY TO TURBO AND CALL SPLICE.COM.
66      * COPY THE OUTPUT .PRN FILE TO THE dBASE III SUBDIRECTORY.
67      *
68      CASE ANS = "1"
69          RUN CD\TURBO
70          RUN SPLICE.COM
71          RUN COPY *.PRN \DBASEIII\*.PRN
72          RUN CD\DBASEIII
73          STORE "1" TO ANS
74      *
75      * CHANGE THE ACTIVE SUBDIRECTORY TO LOTUS AND CALL 123.EXE. THE USER
76      * SUBDIRECTORY WHILE IN LOTUS MUST BE dBASE III.
77      *
78      CASE ANS = "2"
79          RUN CLS
80          RUN ECHO WHEN IN 123, CHANGE THE DEFAULT DIRECTORY TO DBASEIII
81          RUN PAUSE
82          RUN CD\LOTUS
83          RUN 123
84          RUN CD\DBASEIII
85          STORE "2" TO ANS
86      *
87      * CALL THE CONFIGURATION MANAGEMENT SYSTEM dBASE III PROGRAM
88      *
89      CASE ANS = "3"
90          DO MAINMENU
91          STORE "3" TO ANS
92      *
93      * CHANGE THE ACTIVE DIRECTORY TO WORDSTAR AND EDIT THE USER'S MANUAL.
94      *
95      CASE ANS = "4"
96          RUN CLS
97          RUN CD\WORDSTAR
98          RUN COPY USERS.MAN SPLICE.MAN
99          RUN WS.COM SPLICE.MAN
100         RUN DEL SPLICE.MAN
```

```
101      RUN CD\DBASEIII
102      STORE "4" TO ANS
103      *
104      *
105      *      RETURN THE USER TO dBASE SYSTEM CONTROL.
106      *
107      CASE ANS = "5"
108          CLEAR
109          CLEAR ALL
110          EXIT
111      *
112      *      RETURN THE USER TO OPERATING SYSTEM CONTROL.
113      *
114      CASE ANS = "6"
115          CLEAR
116          CLEAR ALL
117          STORE 0 TO CONTINUE
118          QUIT
119      *
120      ENDCASE
121      *
122      *      CONTINUE PROCESSING LOOP CONTROL CHECK.
123      *
124      ENDDO WHILE .T.
125      *****
```

```

1  * PROCEDURE SERNOBLD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : TO BUILD BLANK SERIAL NUMBER RECORDS.
9  *
10 * INPUT FILES   : SERNOTMP.DBF
11 *
12 * OUTPUT FILES  : SERNOTMP.DBF
13 *
14 * MODULES CALLED : NONE
15 *
16 * CALLED BY     : NEWDOADD.PRG
17 *
18 * LOCAL VARIABLES: CTOIQUY, INITIAL, MEFFDATE, MFEATURE, MQTY,
19 *                  MSERIALN, MSITE, REC_COUNT
20 *
21 * DATE LAST TIME MODIFIED =====> 23 DECEMBER 1985 <=====
22 *
23 * IF NOT EOF, DETERMINE THE COMPONENT QUANTITY. WHILE THE QUANTITY
24 * IS GREATER THAN 1, BUILD AND EXPAND A BLANK SERIAL NUMBER RECORD.
25 *
26 STORE 1 TO INITIAL
27 USE SERNOTMP
28 GO TOP
29 DO WHILE .T.
30     IF EOF() = .T. THEN
31         EXIT
32     ELSE
33         *
34         * IF NOT EOF AND NOT A BLANK RECORD, STORE ITEMS TO MEMORY VARIABLES.
35         *
36         IF INITIAL = 1 THEN
37             SET COLOR TO GR+/B, GR+/B
38             @ 17,21 SAY "FEATURE:"
39             @ 17,40 SAY "RECORD NUMBER:"
40             @ 19,18 SAY "Building and expanding sub-record "
41             @ 19,56 SAY "of"
42             STORE 0 TO INITIAL
43         ENDIF
44         SET COLOR TO /BR, /BR
45         @ 17,31 SAY FEATURENO PICT "999999"
46         SET COLOR TO R+/B, R+/B
47         @ 17,55 SAY RECNO() PICT "9999"
48         SET COLOR TO W+/BG, W+/BG
49         STORE 1 TO REC_COUNT
50         @ 19,52 SAY REC_COUNT PICT "999"

```


Page 2

SERNOBLD.PRG Program Listing

```

51      @ 19,59 SAY QTY PICT "999"
52      STORE EFFDATE TO MEFFDATE
53      STORE SITENO TO MSITE
54      STORE FEATURENO TO MFEATURE
55      STORE QTY TO MQTY, CTOTQTY
56      REPLACE TOTQTY WITH MQTY
57      STORE ' ' TO MSERIALN
58      DO WHILE MQTY > 1
59          REC_COUNT = REC_COUNT + 1
60          @ 19,52 SAY REC_COUNT PICT "999"
61          INSERT BLANK
62          REPLACE TOTQTY WITH CTOTQTY
63          REPLACE QTY WITH MQTY - 1
64          REPLACE EFFDATE WITH MEFFDATE
65          REPLACE SITENO WITH MSITE
66          REPLACE FEATURENO WITH MFEATURE
67          REPLACE SERIALNO WITH MSERIALN
68          MQTY = MQTY - 1
69      ENDDO WHILE MQTY > 1
70      *
71      SKIP
72      ENDIF EOF() = .T.
73      *
74      ENDDO WHILE .T.
75      *
76      * CLEAR OUT THE STATUS FIELD LINES
77      *
78      SET COLOR TO W+/B, W+/B
79      @ 15,10 SAY SPACE(60)
80      @ 17,10 SAY SPACE(60)
81      @ 19,10 SAY SPACE(60)
82      *
83      * RETURN TO THE CALLING PROGRAM
84      *
85      RELEASE ALL LIKE M*, CTOTQTY, INITIAL, REC_COUNT
86      CLOSE DATABASES
87      RETURN
88      *****

```

```
1  * PROCEDURE SERNOCMD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER THE OPPORTUNITY TO INPUT
9  *               THE COMPONENT SERIAL NUMBER OR REVIEW THE
10 *               THE SERIAL NUMBER DATABASE RECORDS.
11 *
12 * INPUT FILES   : NONE
13 *
14 * OUTPUT FILES  : NONE
15 *
16 * CALLED BY     : MAINMENU.PRG
17 *
18 * MODULES CALLED : SERNUPD.PRG, SERNOREV.PRG
19 *
20 * LOCAL VARIABLES: SELEKT
21 *
22 * DATE LAST TIME MODIFIED =====> 26 DECEMBER 1985 <=====
23 *
24 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE SELECTION.
25 *
26 STORE "1" TO SELEKT
27 DO WHILE SELEKT < "3"
28     SET COLOR TO W/B, W/B, B
29     CLEAR
30     ?? FLASH + "W.SERNOCMD/"
31     SET CONSOLE OFF
32     WAIT TO SELEKT
33     SET CONSOLE ON
34 *
35 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
36 *
37 DO CASE
38 *
39 *     CALL THE SERIAL NUMBER UPDATE PROGRAM.
40 *     CASE SELEKT = "1"
41 *         DO SERNUPD
42 *
43 *     CALL SERIAL NUMBER REVIEW PROGRAM.
44 *     CASE SELEKT = "2"
45 *         DO SERNOREV
46 *
47 *     RETURN TO THE MAIN MENU PROGRAM.
48 *     CASE SELEKT = "3"
49 *
50 ENDCASE
```

Page 2

SERNOCMD.PRG Program Listing

```
51  *  
52  ENDDO (WHILE SELEKT < "3")  
53  *  
54  * RETURN TO THE CALLING PROGRAM  
55  *  
56  RETURN  
57  *****
```

Page 1

SERNOREV.PRG Program Listing

```

1  * PROCEDURE SERNOREV.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              LCDR WINSTON H. BUCKLEY, SC, USN
5  *              LCDR ROBERT F. BRADO, USN
6  *              LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : TO ENABLE THE USER TO REVIEW ALL RECORDS IN THE
9  *              SERIAL NUMBER DATABASE.
10 *
11 * INPUT FILES   : SERIALNO.DBF, SERNOSIT.NDX
12 *
13 * OUTPUT FILES  : SERIALNO.DBF, SERNOSIT.NDX
14 *
15 * CALLED BY     : SERNOCMD.PRG
16 *
17 * MODULES CALLED : DELAY.PRG
18 *
19 * GLOBAL VARIABLE: HIFNUM, HISITE, LOFNUM, LOSITE
20 *
21 * LOCAL VARIABLES: ACCEPT, CHOICE, CURRENTNO, EOF, FIRST_REC, LAST_REC,
22 *                 MFEAT, MFEATURE, MSITE, TOF
23 *
24 * DATE LAST TIME MODIFIED =====> 26 DECEMBER 1985 <=====
25 *
26 * CASE SELECTION = 2      REVIEW SERIAL NUMBER FILE RECORDS
27 *
28 * USE SERIAL NUMBER DATABASE INDEXED ON SITE NUMBER AND WAIT FOR THE
29 * USER TO INPUT THE SITE NUMBER, THEN START REVIEWING FROM THAT POINT.
30 *
31 SET ESCAPE OFF
32 SET TALK OFF
33 SELECT 1
34 USE SERIALNO
35 GO TOP
36 SET COLOR TO W+/B, W+/B, B
37 CLEAR
38 IF EOF() = .T. THEN
39     SET COLOR TO W+/R, W+/R
40     @ 13,22 SAY " The SERIAL NUMBER Database is EMPTY! "
41     DO DELAY
42     RETURN
43 ENDIF
44 ?? FLASH + "S.SERIALNO.SCR/"
45 SET COLOR TO W+/B, W+/B
46 @ 24,0 SAY SPACE(80)
47 SET COLOR TO R+/ , R+/
48 @ 3,26 SAY " SERIAL NUMBER REVIEW FORMAT "
49 STORE " Enter 00 to start at TOF, 99 to start at EOF or a site number " +
50     "between " + LOSITE + " and " + HISITE + " " TO MESSAGE

```

AD-A168 517

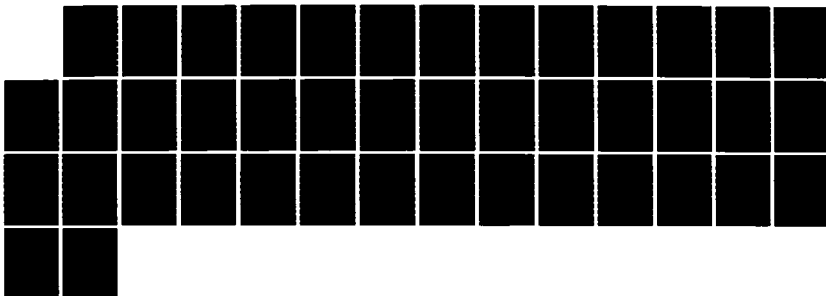
DEVELOPMENT OF AN AUTOMATED MICROCOMPUTER
KNOWLEDGE-BASED INTEGRATED CON. (U) NAVAL POSTGRADUATE
SCHOOL MONTEREY CA R L BEARD MAR 86

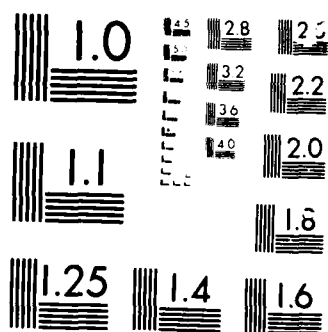
5/3

UNCLASSIFIED

F/G 9/2

NL





U.S. GOVERNMENT PRINTING OFFICE

```

51  *
52  DO WHILE .T.
53      SET COLOR TO /W, /W
54      @ 24,0 SAY MESSAGE
55      SET COLOR TO /BR, /BR
56      STORE '00' TO MSITE
57      @ 09,20 GET MSITE PICT '99'
58      READ
59      IF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR.;
60          MSITE = '99') THEN
61          SET COLOR TO W/B, W/B
62          @ 24,0 SAY SPACE(80)
63          SET COLOR TO W+/R,W+/R
64          STORE ' Response must be between ' + LOSITE + ' and ' +;
65              HISITE + ', Zero (00) or 99 ' TO ERROR
66          @ 24,13 SAY ERROR
67          DO DELAY
68          LOOP
69      ELSE
70          IF (MSITE = '00' .OR. MSITE = '99') THEN
71              IF MSITE = '00' THEN
72                  GO BOTTOM
73                  STORE RECNO() TO LAST_REC
74                  GO TOP
75                  STORE RECNO() TO FIRST_REC
76              ELSE
77                  GO TOP
78                  STORE RECNO() TO FIRST_REC
79                  GO BOTTOM
80                  STORE RECNO() TO LAST_REC
81              ENDIF MSITE = '00'
82              EXIT
83          ELSE
84              USE SERIALNO INDEX SERNOSIT
85              GO TOP
86              FIND &MSITE
87              IF EOF() = .T. THEN
88                  SET COLOR TO W/B, W/B
89                  @ 24,0 SAY SPACE(80)
90                  SET COLOR TO W+/R, W+/R
91                  STORE " No records exist for site number " + MSITE +;
92                      ", try again " TO ERROR
93                  @ 24,16 SAY ERROR
94                  DO DELAY
95                  LOOP
96              ELSE
97                  EXIT
98              ENDIF EOF() = .T.
99          ENDIF (MSITE = '00' .OR. MSITE = '99')
100      ENDIF .NOT. ((MSITE >= '00' .AND. MSITE <= HISITE) .OR. MSITE = '99')

```

```

101 ENDDO WHILE .T.
102 *
103 STORE SPACE(10) + 'Enter "00  " to start at TOF or a six digit ' +;
104     'feature number' + SPACE(10) TO MESSAGE
105 IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
106     DO WHILE .T.
107         SET COLOR TO /W, /W
108         @ 24,0 SAY MESSAGE
109         SET COLOR TO /BR, /BR
110         STORE '00  ' TO MFEAT
111         @ 13,45 GET MFEAT PICT '999999'
112         READ
113         IF .NOT. ((MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) .OR.;
114             MFEAT = '00  ') THEN
115             SET COLOR TO W/B, W/B
116             @ 24,0 SAY SPACE(80)
117             SET COLOR TO W+/R, W+/R
118             STORE ' Response must be between ' + LOFNUM + ' and ' +;
119                 HIFNUM + ' or Zero (00) ' TO ERROR
120             @ 24,9 SAY ERROR
121             DO DELAY
122             LOOP
123         ELSE
124             IF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
125                 IF MFEAT = '99  ' THEN
126                     SET COLOR TO W/B, W/B
127                     @ 24,0 SAY SPACE(80)
128                     SET COLOR TO W+/R, W+/R
129                     STORE ' Response must be between ' + LOFNUM +;
130                         ' and ' + HIFNUM + ' or Zero (00) ' TO ERROR
131                     @ 24,9 SAY ERROR
132                     DO DELAY
133                     LOOP
134                 ENDIF MFEAT = '99  '
135                 STORE MSITE + MFEAT TO MKEY
136                 USE SERIALNO INDEX SERNOFEA
137                 GO TOP
138                 FIND &MKEY
139                 IF EOF() = .T. THEN
140                     SET COLOR TO W/B, W/B
141                     @ 24,0 SAY SPACE(80)
142                     SET COLOR TO W+/R, W+/R
143                     STORE " No record with feature number " + MFEAT +;
144                         " exists, try again " TO ERROR
145                     @ 24,12 SAY ERROR
146                     DO DELAY
147                     LOOP
148                 ELSE
149                     EXIT
150                 ENDIF EOF() = .T.

```


Page 4

SERNOREV.PRG Program Listing

```
151         ELSE
152             EXIT
153         ENDIF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
154     ENDIF
155     ENDDO WHILE .T.
156 ENDIF .NOT. (MSITE >= LOSITE .OR. MSITE <= HISITE)
157 *
158 STORE " At beginning of records for site number " +;
159     MSITE + " " TO TOF
160 STORE " At end of records for site number " + MSITE + " " TO EOF
161 SET COLOR TO W/B, W/B
162 @ 24,0 SAY SPACE(80)
163 *
164 DO WHILE .T.
165 *
166 * USING THE SERIAL NUMBER REVIEW FORMAT FILE TO PRODUCE THE SCREEN
167 * DISPLAY, IF NOT AT THE END OF FILE.
168 *
169     STORE FEATURENO TO MFEATURE
170     SELECT 2
171     USE DESCRIP INDEX DESCRIP
172     FIND &MFEATURE
173     STORE CLIN TO MCLIN
174     STORE DESCRIPT TO MDESCRIPT
175     SELECT 1
176     SET COLOR TO R+/B, R+/B
177     @ 6,45 SAY RECNO() PICT "9999"
178     SET COLOR TO /BR, /BR
179     @ 9,20 SAY SITENO PICT "99"
180     @ 9,68 SAY EFFDATE PICT "999999"
181     @ 12,45 SAY MCLIN PICT "9999"
182     @ 13,45 SAY FEATURENO PICT "999999"
183     @ 14,45 SAY MDESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
184     @ 15,45 SAY TOTQTY PICT "999"
185     SET COLOR TO W+/BG, W+/BG
186     @ 17,45 SAY QTY PICT "999"
187     @ 17,52 SAY TOTQTY PICT "999"
188     SET COLOR TO /BR, /BR
189     @ 19,45 SAY SERIALNO PICT "!!!!!!!!!!"
190 ENDIF
191 *
192     SET COLOR TO R+/B, R+/B
193     STORE "N" TO CHOICE
194     @ 22,68 GET CHOICE PICT "!"
195     READ
196 *
197 * ENSURE THAT THE USER'S PROMPT IS EITHER "N", "P" OR "X"
198 *
199     DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
200         IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
```

```
201      SET COLOR TO W+/R, W+/R
202      @ 24,23 SAY " Response must be either N, P or X "
203      DO DELAY
204      STORE "N" TO CHOICE
205  ENDIF
206      SET COLOR TO R+/B, R+/B
207      @ 22,68 GET CHOICE PICT "!"
208      READ
209  ENDDO
210  *
211      SET COLOR TO W+/R, W+/R
212  *
213  * SKIP TO THE NEXT RECORD TO BE REVIEWED
214  *
215      IF CHOICE = "N" THEN
216          IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
217              SKIP
218              IF EOF() = .T. THEN
219                  SKIP - 1
220                  @ 24,21 SAY EOF
221                  DO DELAY
222                  LOOP
223              ELSE
224                  IF .NOT. (SITENO = MSITE) THEN
225                      SKIP - 1
226                      @ 24,21 SAY EOF
227                      DO DELAY
228                      LOOP
229                  ENDIF
230              ENDIF EOF() = .T.
231          ELSE
232              IF RECNO() = LAST_REC THEN
233                  GO TOP
234              ELSE
235                  SKIP
236              ENDIF
237          ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
238      ENDIF CHOICE = "N"
239  *
240  * SKIP TO THE PREVIOUS RECORD
241  *
242      IF CHOICE = "P" THEN
243          STORE RECNO() TO CURRENINO
244          IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
245              SKIP - 1
246              IF BOF() = .T. THEN
247                  GOTO CURRENINO
248                  @ 24,16 SAY TOP
249              DO DELAY
250              LOOP
```

```
251         ELSE
252             IF .NOT. (SITE NO = MSITE) THEN
253                 SKIP
254                 @ 24,16 SAY TOF
255                 DO DELAY
256                 LOOP
257             ENDIF
258         ENDIF BOF() = .T.
259     ELSE
260         IF RECNO() = FIRST_REC THEN
261             GO BOTTOM
262         ELSE
263             SKIP - 1
264         ENDIF
265     ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE')
266     ENDIF CHOICE = "p"
267 *
268 * USER HAS DECIDED TO EXIT THE REVIEW
269 *
270     IF CHOICE = "X"
271         EXIT
272     ENDIF
273 ENDDO WHILE .T.
274 *
275 * RETURN TO CALLING PROGRAM.
276 *
277 RELEASE ALL LIKE M*, ACCEPT, CHOICE, CURRENTNO, EOF, FIRST_REC,;
278     LAST_REC, TOF
279 CLOSE DATABASES
280 RETURN
281 *****
```

```

1  * PROCEDURE SERNOUPD.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *               LCDR WINSTON H. BUCKLEY, SC, USN
5  *               LCDR ROBERT F. BRADO, USN
6  *               LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : TO ENABLE THE USER TO INPUT THE SERIAL NUMBERS FOR
9  *               THE SERIAL NUMBER DATABASE.
10 *
11 * INPUT FILES   : SERIALNO.DBF, SERNOSIT.NDX
12 *
13 * OUTPUT FILES  : SERIALNO.DBF, SERNOSIT.NDX
14 *
15 * CALLED BY     : SERNOCMD.PRG
16 *
17 * MODULES CALLED : DELAY.PRG
18 *
19 * GLOBAL VARIABLE: HDATE, HIFNUM, HISITE, LDATE, LOFNUM, LOSITE
20 *
21 * LOCAL VARIABLES: ACCEPT, ANS, CHOICE, CURRENTNO, EOF, INTRO, MDATE,;
22 *                 MDAY, MESSAGE, MMONTH, MOLDATE, MYEAR, NODATE,;
23 *                 NOFIND, SYSDATE, TOF
24 *
25 * DATE LAST TIME MODIFIED =====> 26 DECEMBER 1985 <=====
26 *
27 * CASE SELECTION = 2      REVIEW SERIAL NUMBER FILE RECORDS
28 *
29 * USE SERIAL NUMBER DATABASE INDEXED ON SITE NUMBER AND WAIT FOR THE
30 * USER TO INPUT THE SITE NUMBER, THEN START REVIEWING FROM THAT POINT.
31 *
32 SET ESCAPE OFF
33 SET TALK OFF
34 SELECT 1
35 USE SERIALNO
36 GO TOP
37 SET COLOR TO W+/B, W+/B, B
38 CLEAR
39 IF EOF() = .T. THEN
40     SET COLOR TO W+/R, W+/R
41     @ 13,22 SAY " The SERIAL NUMBER Database is EMPTY! "
42     DO DELAY
43     RETURN
44 ENDIF
45 ?? FLASH + "S.SERIALNO.SCR/"
46 SET COLOR TO W+/B, W+/B
47 @ 24,0 SAY SPACE(80)
48 SET COLOR TO R+/ , R+/
49 @ 3,26 SAY " SERIAL NUMBER UPDATE FORMAT "
50 STORE SPACE(22) + "Enter a Site Number between " + LOSITE +;

```

```

51      " and " + HISITE + SPACE(21) TO MESSAGE
52      USE SERIALNO INDEX SERNOSIT
53      *
54      DO WHILE .T.
55          SET COLOR TO /W, /W
56          @ 24,0 SAY MESSAGE
57          SET COLOR TO /BR, /BR
58          STORE LOSITE TO MSITE
59          @ 29,20 GET MSITE PICT '99'
60          READ
61          IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
62              SET COLOR TO W/B, W/B
63              @ 24,0 SAY SPACE(80)
64              SET COLOR TO W+/R, W+/R
65              STORE ' Response must be between ' + LOSITE + ;
66                  ' and ' + HISITE + ' ' TO ERROR
67              @ 24,22 SAY ERROR
68              DO DELAY
69              LOOP
70          ELSE
71              GO TOP
72              FIND &MSITE
73              IF EOF() = .T. THEN
74                  SET COLOR TO W/B, W/B
75                  @ 24,0 SAY SPACE(80)
76                  SET COLOR TO W+/R, W+/R
77                  STORE " No records exist for site number " + MSITE + ;
78                      ", try again " TO ERROR
79                  @ 24,16 SAY ERROR
80                  DO DELAY
81                  LOOP
82              ELSE
83                  EXIT
84              ENDIF EOF() = .T.
85          ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
86      ENDDO WHILE .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
87      *
88      STORE SPACE(17) + 'Input Effective Date (Range ' + LODATE + ;
89          ' to ' + HIDATE + ' )' + SPACE(17) TO MESSAGE
90      STORE DLOC(DATE()) TO SYSDATE
91      STORE SUBSTR(SYSDATE,7,2) + SUBSTR(SYSDATE,1,2) + ;
92          SUBSTR(SYSDATE,4,2) TO MDATE
93      STORE 0 TO NOFIND
94      STORE "000000" TO MOLDATE
95      USE SERIALNO INDEX SERNODAT
96      *
97      DO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDATE)
98          SET COLOR TO /W, /W
99          @ 24,0 SAY MESSAGE
100         STORE MDATE TO MOLDATE

```

Page 3

SERNOUPD.PRG Program Listing

```

101 SET COLOR TO /BR, /BR
102 @ 9,68 GET MOLDATE PICT "999999"
103 READ
104 DO WHILE .T.
105     IF .NOT. (SUBSTR(MOLDATE,1,2) > "83" .AND.;
106             SUBSTR(MOLDATE,1,2) <= "99") THEN
107         SET COLOR TO W/B, W/B
108         @ 24,0 SAY SPACE(80)
109         SET COLOR TO W+/R, W+/R
110         STORE " Year portion of date must be between 84 and 99 ";
111         TO ERROR
112         @ 24,16 SAY ERROR
113         DO DELAY
114         SET COLOR TO /W, /W
115         @ 24,0 SAY MESSAGE
116         STORE SUBSTR(MDATE,1,2) TO MYEAR
117         SET COLOR TO /BR, /BR
118         @ 9,68 GET MYEAR PICT "99"
119         READ
120         STORE MYEAR + SUBSTR(MOLDATE,3,4) TO MOLDATE
121         LOOP
122     ELSE
123         EXIT
124     ENDIF
125 ENDDO WHILE .T.
126 *
127 DO WHILE .T.
128     IF .NOT. (SUBSTR(MOLDATE,3,2) >= "01" .AND.;
129             SUBSTR(MOLDATE,3,2) <= "12") THEN
130         SET COLOR TO W/B, W/B
131         @ 24,0 SAY SPACE(80)
132         SET COLOR TO W+/R, W+/R
133         @ 24,16 SAY " Month portion of date must be between 01 and 12 "
134         DO DELAY
135         SET COLOR TO /W, /W
136         @ 24,0 SAY MESSAGE
137         STORE SUBSTR(MDATE,3,2) TO MMONTH
138         SET COLOR TO /BR, /BR
139         @ 9,70 GET MMONTH PICT "99"
140         READ
141         STORE SUBSTR(MOLDATE,1,2) + MMONTH +;
142             SUBSTR(MOLDATE,5,2) TO MOLDATE
143         LOOP
144     ELSE
145         EXIT
146     ENDIF
147 ENDDO WHILE .T.
148 *
149 DO WHILE .T.
150     IF ((SUBSTR(MOLDATE,3,2)="04" .OR. SUBSTR(MOLDATE,3,2)="06" .P.;

```

```

151 SUBSTR(MOLDATE,3,2)="09" .OR. SUBSTR(MOLDATE,3,2)="11" ) .AND.;
152 .NOT. (SUBSTR(MOLDATE,5,2)>="01" .AND.;
153 SUBSTR(MOLDATE,5,2)<="30")) THEN
154 SET COLOR TO W/B, W/B
155 @ 24,0 SAY SPACE(80)
156 SET COLOR TO W+/R, W+/R
157 @ 24,16 SAY " Day portion of date must be between 01 and 30 "
158 DO DELAY
159 SET COLOR TO /W, /W
160 @ 24,0 SAY MESSAGE
161 STORE SUBSTR(MDATE,5,2) TO MDAY
162 SET COLOR TO /BR, /BR
163 @ 9,72 GET MDAY PICT "99"
164 READ
165 STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
166 LOOP
167 ELSE
168 *
169 IF (SUBSTR(MOLDATE,3,2) = "02" .AND. .NOT.;
170 (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
171 SUBSTR(MOLDATE,5,2) <= "28")) THEN
172 SET COLOR TO W/B, W/B
173 @ 24,0 SAY SPACE(80)
174 SET COLOR TO W+/R, W+/R
175 @ 24,16 SAY " Day portion of date must be between 01 and 28 "
176 DO DELAY
177 SET COLOR TO /W, /W
178 @ 24,0 SAY MESSAGE
179 STORE SUBSTR(MDATE,5,2) TO MDAY
180 SET COLOR TO /BR, /BR
181 @ 9,72 GET MDAY PICT "99"
182 READ
183 STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
184 LOOP
185 ELSE
186 *
187 IF .NOT. (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
188 SUBSTR(MOLDATE,5,2) <= "31") THEN
189 SET COLOR TO W/B, W/B
190 @ 24,0 SAY SPACE(80)
191 SET COLOR TO W+/R, W+/R
192 @ 24,16 SAY " Day portion of date must be between 01 and 31 "
193 DO DELAY
194 SET COLOR TO /W, /W
195 @ 24,0 SAY MESSAGE
196 STORE SUBSTR(MDATE,5,2) TO MDAY
197 SET COLOR TO /BR, /BR
198 @ 9,72 GET MDAY PICT "99"
199 READ
200 STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE

```

```
201         LOOP
202     ELSE
203         EXIT
204     ENDIF
205     ENDIF
206     ENDIF
207     ENDDO WHILE .T.
208 *
209 * SEE IF THE USER'S DATE IS A VALID DATE FOR THE SITE SELECTED
210 *
211     STORE MSITE + MOLDATE TO MKEY
212     GO TOP
213     FIND &MKEY
214     IF EOF() = .T. THEN
215         NOFIND = NOFIND + 1
216         IF NOFIND = 3 THEN
217             SET COLOR TO W+/B, W+/B
218             @ 24,0 SAY SPACE(80)
219             ?? FLASH + "W.SERNOFND/"
220             SET CONSOLE OFF
221             WAIT TO ANS
222             SET CONSOLE ON
223             IF ANS = "2" THEN
224                 RELEASE ALL LIKE M*, ACCEPT, ANS, CHOICE,;
225                     CURRENINO, EOF, INTRO, NODATE,;
226                     NOFIND, SYSDATE, TOF
227                 CLOSE DATABASES
228                 RETURN
229             ELSE
230                 SET COLOR TO /W, /W
231                 @ 24,0 SAY MESSAGE
232                 STORE 0 TO NOFIND
233                 STORE '000000' TO MOLDATE
234                 LOOP
235             ENDIF ANS = "2"
236         ELSE
237             SET COLOR TO W/B, W/B
238             @ 24,0 SAY SPACE(80)
239             STORE "EFFECTIVE DATE " + MOLDATE + " does not exist for site " +;
240                 MSITE + ", try another " TO NODATE
241             SET COLOR TO W+/R, W+/R
242             @ 24,10 SAY NODATE
243             DO DELAY
244             SET COLOR TO /W, /W
245             @ 24,0 SAY MESSAGE
246             STORE "000000" TO MOLDATE
247             LOOP
248         ENDIF NOFIND = 3
249     ENDIF EOF() = .T.
250     ENDDO WHILE .NOT. (MOLDATE >= IODATE .AND. MOLDATE <= HIDATE)
```



```

251 *
252 STORE SPACE(10) + 'Enter a six digit feature number between ' + LOFNUM +;
253 ' and ' + HIFNUM + SPACE(11) TO MESSAGE
254 SET COLOR TO /W, /W
255 @ 24,0 SAY MESSAGE
256 STORE '999999' TO MFEAT
257 STORE 0 TO NOFIND
258 *
259 DO WHILE .T.
260     DO WHILE .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
261         SET COLOR TO /BR, /BR
262         STORE '010201' TO MFEAT
263         @ 13,45 GET MFEAT PICT '999999'
264         READ
265         IF .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) THEN
266             SET COLOR TO W/B, W/B
267             @ 24,0 SAY SPACE(80)
268             SET COLOR TO W+/R, W+/R
269             STORE 'Response must be between ' + LOFNUM +;
270             ' and ' + HIFNUM + ' ' TO ERROR
271             @ 24,17 SAY ERROR
272             DO DELAY
273             SET COLOR TO /W, /W
274             @ 24,0 SAY MESSAGE
275             LOOP
276     ELSE
277         IF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM) THEN
278             USE DESCRIP INDEX DESCRIP
279             GO TOP
280             FIND &MFEAT
281             IF EOF() = .T. THEN
282                 NOFIND = NOFIND + 1
283                 IF NOFIND = 3 THEN
284                     SET COLOR TO W+/B, W+/B
285                     @ 24,0 SAY SPACE(80)
286                     ?? FLASH + "W.SERNOFND/"
287                     SET CONSOLE OFF
288                     WAIT TO ANS
289                     SET CONSOLE ON
290                     IF ANS = "2" THEN
291                         RELEASE ALL LIKE M*, ACCEPT, ANS, CHOICE,;
292                         CURRENINO, EOF, INITIO, NODATE,;
293                         NOFIND, SYSDATE, TOF
294                     CLOSE DATABASES
295                     RETURN
296             ELSE
297                 SET COLOR TO /W, /W
298                 @ 24,0 SAY MESSAGE
299                 STORE 0 TO NOFIND
300                 STORE '999999' TO MFEAT

```

```

301         LOOP
302         ENDIF ANS = "2"
303     ELSE
304         SET COLOR TO W/B, W/B
305         @ 24,0 SAY SPACE(80)
306         SET COLOR TO W+/R, W+/R
307         STORE " No record exists for feature number " +;
308             MFEAT + ", try again " TO ERROR
309         @ 24,12 SAY ERROR
310         DO DELAY
311         SET COLOR TO /W, /W
312         @ 24,0 SAY MESSAGE
313         STORE '999999' TO MFEAT
314         LOOP
315     ENDIF NOFIND = 3
316     ENDIF EOF() = .T.
317     ENDIF (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
318     ENDIF .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
319     ENDDO WHILE .NOT. (MFEAT >= LOFNUM .AND. MFEAT <= HIFNUM)
320 *
321     STORE MOLDATE + MSITE + MFEAT TO MKEY
322     USE SERIALNO INDEX SERNOPRJ
323     GO TOP
324     FIND &MKEY
325     IF EOF() = .T. THEN
326         SET COLOR TO W/B, W/B
327         @ 24,0 SAY SPACE(80)
328         SET COLOR TO W+/R, W+/R
329         STORE " Feature number " + MFEAT + " for site " + MSITE +;
330             " on date " + MOLDATE +;
331             " does not exist, try again " TO ERROR
332         @ 24,0 SAY ERROR
333         DO DELAY
334         SET COLOR TO W+/B, W+/B
335         ?? FLASH + "W.SERNOEND/"
336         SET CONSOLE OFF
337         WAIT TO ANS
338         SET CONSOLE ON
339         IF ANS = "2" THEN
340             RELEASE ALL LIKE M*, ACCEPT, ANS, CHOICE, CURRENTNO, EOF,;
341                 INTRO, NODATE, NOFIND, SYSDATE, TOP
342             CLOSE DATABASES
343             RETURN
344         ELSE
345             SET COLOR TO W/B, W/B
346             @ 21,10 SAY SPACE(60)
347             SET COLOR TO /W, /W
348             @ 24,0 SAY MESSAGE
349             STORE '999999' TO MFEAT
350             LOOP

```

```

351     ENDIF ANS = "2"
352     ELSE
353         EXIT
354     ENDIF EOF() = .T.
355 ENDDO WHILE .T.
356 *
357 STORE " At beginning of records for site number " +;
358     MSITE + " " TO TOF
359 STORE " At end of records for site number " + MSITE + " " TO EOF
360 SET COLOR TO W/B, W/B
361 @ 24,0 SAY SPACE(80)
362 *
363 STORE SPACE(16) + 'Press "Page Down" key to terminate record update' +;
364     SPACE(16) TO MESSAGE
365 STORE 1 TO INTRO
366 DO WHILE .T.
367     SET COLOR TO /W, /W
368     @ 24,0 SAY MESSAGE
369 *
370 * USING THE SERIAL NUMBER UPDATE FORMAT FILE TO PRODUCE THE SCREEN
371 * DISPLAY, IF NOT AT THE END OF FILE.
372 *
373     STORE SERIALNO TO MSERIAL
374     STORE FEATURENO TO MFEAT
375 *
376 * INFORM THE USER OF HOW TO TERMINATE THE UPDATE OF A RECORD
377 *
378     IF INTRO = 1 THEN
379         STORE 0 TO INTRO
380         ?? FLASH + "W.SERNOUPD/"
381         SET CONSOLE OFF
382         WAIT TO ANS
383         SET CONSOLE ON
384     ENDIF
385 *
386     SELECT 2
387     USE DESCRIP INDEX DESCRIP
388     FIND &MFEAT
389     STORE CLIN TO MCLIN
390     STORE DESCRIPT TO MDESCRIP
391     SELECT 1
392     SET COLOR TO R+/B, R+/B
393     @ 6,45 SAY RECNO() PICT "9999"
394     SET COLOR TO /BR, /BR
395     @ 9,20 SAY SITENO PICT "99"
396     @ 9,68 SAY EFFDATE PICT "999999"
397     @ 12,45 SAY MCLIN PICT "9999"
398     @ 13,45 SAY MFEAT PICT "999999"
399     @ 14,45 SAY MDESCRIP PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
400     @ 15,45 SAY TOTQTY PICT "999"

```

```
401 SET COLOR TO W+/BG, W+/BG
402 @ 17,45 SAY QTY PICT "999"
403 @ 17,52 SAY TOTQTY PICT "999"
404 SET COLOR TO /BR, /BR
405 @ 19,45 GET MSERIAL PICT "!!!!!!!"
406 READ
407 SET COLOR TO W/B, W/B
408 @ 24,0 SAY SPACE(80)
409 *
410 IF .NOT. (SERIALNO = MSERIAL) THEN
411 *
412 *     ASK THE USER IF HE/SHE DESIRES TO ACCEPT THE CHANGES
413 *
414     SET COLOR TO W+/B, W+/B
415     @ 21,12 SAY "Do you want to accept the change? (Yes or No):"
416     SET COLOR TO R+/B, R+/B
417     @ 21,48 SAY "Y"
418     @ 21,55 SAY "N"
419     STORE "N" TO ACCEPT
420     @ 21,61 GET ACCEPT PICT "!"
421     READ
422 *
423 *     ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
424 *
425     DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
426         IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
427             SET COLOR TO W+/R, W+/R
428             @ 24,24 SAY " Response must be either N or Y "
429             DO DELAY
430             STORE "N" TO ACCEPT
431         ENDIF
432         SET COLOR TO R+/B, R+/B
433         @ 21,61 GET ACCEPT PICT "!"
434         READ
435     ENDDO
436     SET COLOR TO W/B, W/B
437     @ 21,10 SAY SPACE(55)
438 *
439 *     STORE THE CHANGED EDIT FIELD FROM THE WORK AREA INTO THE
440 *     DATABASE VARIABLE
441 *
442     IF ACCEPT = "Y" THEN
443         REPLACE SERIALNO WITH MSERIAL
444     ELSE
445         SET COLOR TO /BR, /BR
446         @ 19,45 SAY SERIALNO PICT "!!!!!!!"
447     ENDIF ACCEPT = "Y"
448 ENDIF .NOT. (SERIALNO = MSERIAL)
449 *
450 SET COLOR TO R+/B, R+/B
```

```
451 STORE "N" TO CHOICE
452 @ 22,68 GET CHOICE PICT "!"
453 READ
454 *
455 * ENSURE THAT THE USER'S RESPONSE IS EITHER "N", "P" OR "X"
456 *
457 DO WHILE .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X")
458     IF .NOT. (CHOICE = "N" .OR. CHOICE = "P" .OR. CHOICE = "X") THEN
459         SET COLOR TO W+/R, W+/R
460         @ 24,23 SAY " Response must be either N, P or X "
461         DO DELAY
462         STORE "N" TO CHOICE
463     ENDIF
464     SET COLOR TO R+/B, R+/B
465     @ 22,68 GET CHOICE PICT "!"
466     READ
467 ENDDO
468 *
469 * SKIP TO THE NEXT RECORD TO BE REVIEWED
470 *
471 IF CHOICE = "N" THEN
472     IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
473         SKIP
474         IF EOF() = .T. THEN
475             SKIP - 1
476             SET COLOR TO W+/R, W+/R
477             @ 24,21 SAY EOF
478             DO DELAY
479         ELSE
480             IF .NOT. (SITENO = MSITE) THEN
481                 SKIP - 1
482                 SET COLOR TO W+/R, W+/R
483                 @ 24,21 SAY EOF
484                 DO DELAY
485             ENDIF
486             ENDIF EOF() = .T.
487             ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
488             ENDIF CHOICE = "N"
489 *
490 * SKIP TO THE PREVIOUS RECORD
491 *
492 IF CHOICE = "P" THEN
493     STORE RECNO() TO CURRENTNO
494     IF (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
495         SKIP - 1
496         IF BOF() = .T. THEN
497             GOTO CURRENTNO
498             SET COLOR TO W+/R, W+/R
499             @ 24,16 SAY EOF
500             DO DELAY
```

```
501         ELSE
502             IF .NOT. (SITE NO = MSITE) THEN
503                 SKIP
504                 SET COLOR TO W+/R, W+/R
505                 @ 24,16 SAY TOF
506                 DO DELAY
507             ENDIF
508             ENDIF BOF() = .T.
509             ENDIF (MSITE >= LOSITE .AND. MSITE <= HISITE)
510             ENDIF CHOICE = "P"
511             * USER HAS DECIDED TO EXIT THE REVIEW
512             *
513             IF CHOICE = "X"
514                 EXIT
515             ENDIF
516         ENDDO WHILE .T.
517         *
518         * RETURN TO CALLING PROGRAM.
519         *
520         RELEASE ALL LIKE M*, ACCEPT, ANS, CHOICE, CURRENTNO, EOF, INTRO,;
521             NODATE, NOFIND, SYSDATE, TOF
522         CLOSE DATABASES
523         RETURN
524         *****
```

```
1 * PROCEDURE SITERPTS.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              : LCDR WINSTON H. BUCKLEY, SC, USN
5 *              : LCDR ROBERT F. BRADO, USN
6 *              : LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : PROVIDE THE USER A SELECTION OF SITE LEVEL REPORTS.
9 *
10 * INPUT FILES  : NONE.
11 *
12 * OUTPUT FILES : NONE.
13 *
14 * CALLED BY    : REPORCMD.PRG
15 *
16 * MODULES CALLED : EQPSTRPT.PRG, MNLSTRPT.PRG, SNOSTRPT.PRG
17 *
18 * LOCAL VARIABLES: SELEKT
19 *
20 * DATE LAST TIME MODIFIED =====> 18 DECEMBER 1985 <=====
21 *
22 * DISPLAY THE PROCESS MENU TO THE USER AND WAIT FOR THE SELECTION.
23 *
24 STORE "1" TO SITERPTS
25 DO WHILE SITERPTS < "4"
26     SET COLOR TO W/B, W/B, B
27     CLEAR
28     ?? FLASH + "W.SITERPTS/"
29     SET CONSOLE OFF
30     WAIT TO SITERPTS
31     SET CONSOLE ON
32 *
33 * PROCESS ROUTINE BASED ON THE USER'S SELECTION.
34 *
35 DO CASE
36 *
37 *     CALL THE EQUIPMENT SITE LEVEL REPORT.
38 *     CASE SITERPTS = "1"
39 *         DO EQPSTRPT
40 *
41 *     CALL THE MANUAL SITE LEVEL REPORT.
42 *     CASE SITERPTS = "2"
43 *         DO MNLSTRPT
44 *
45 *     CALL THE SERIAL NUMBER SITE LEVEL REPORT.
46 *     CASE SITERPTS = "3"
47 *         DO SNOSTRPT
48 *
49 *     RETURN TO THE SPLICE REPORTING LEVEL MENU.
50 *     CASE SITERPTS = "4"
```

Page 2

SITERPTS.PRG Program Listing

```

51  *
52  ENDCASE
53  *
54  ENDDO (WHILE SITERPTS = "4")
55  *
56  * RETURN TO THE CALLING PROGRAM
57  *
58  RETURN
59  *****

```


Page 1

SNODTRPT.PRG Program Listing

```

1  * PROCEDURE SNODTRPT.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              : LCDR WINSTON H. BUCKLEY, SC, USN
5  *              : LCDR ROBERT F. BRADO, USN
6  *              : LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER A SPLICE SERIAL NUMBER
9  *              : EFFECTIVE DELIVERY ORDER LEVEL REPORT.
10 *
11 * INPUT FILES   : SERIALNO.DBF, SERNODAT.NDX, DESCRIP.DBF,
12 *              : DESCRIP.NDX, EQUIP.DBF, EQUIPSIT.NDX
13 *
14 * CALLED BY     : DATERTPTS.PRG
15 *
16 * MODULES CALLED : NONE.
17 *
18 * GLOBAL VARIABLE: HDATE, HISITE, LDATE, LOSITE
19 *
20 * LOCAL VARIABLES: ACCEPT, CHOICE, COLCNT, ERROR, LINECT, MDAY, MKEY,
21 *              : MMONTH, MNEWDATE, MOLDATE, MSITE, MYEAR, PAGENO,
22 *              : SYSDATE, TODAY, TODATE
23 *
24 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
25 *
26 * CASE SELECTION = 3   SERIAL NUMBER EFFECTIVE DELIVERY ORDER LEVEL REPORT
27 *
28 * CALL THE SERIAL NUMBER DATABASE INDEXED ON SITE NUMBER.  DISPLAY
29 * THE EFFECTIVE DELIVERY ORDER DATES FOR THE USER TO SELECT FROM.
30 * CALL SERIAL NUMBER DATABASE INDEXED ON EFFECTIVE DELIVERY ORDER DATE
31 * AND SITE NUMBER. COPY TO TEMPONE, INDEXED ON FEATURE NUMBER. RELATE
32 * TO THE DESCRIPTION FILE AND PRODUCE REPORT.
33 *
34 SET ESCAPE OFF
35 SET TALK OFF
36 SET COLOR TO W+/B, W+/B, B
37 CLEAR
38 USE SERIALNO
39 GO TOP
40 IF DEF() = .T. THEN
41     SET COLOR TO W+/R, W+/R
42     @ 13,22 SAY " The SERIAL NUMBER Database is EMPTY! "
43     DO DELAY
44     RETURN
45 ENDIF
46 ?? FLASH + "S.REPORTS.SCR/"
47 @ 24,0 SAY SPACE(80)
48 SET COLOR TO R+/ , R+/
49 @ 2,26 SAY " SITE SERIAL NUMBER REPORT "
50 SET COLOR TO W+/BR, W+/BR

```

```

51 @ 13,15 SAY "Enter site number for which the report is desired:"
52 *
53 * ENSURE THAT TEMPORARY DATABASE AND INDEX DO NOT EXIST,
54 * IF SO ERASE THEM
55 *
56 SET CONSOLE OFF
57 ERASE TEMPONE.DBF
58 ERASE TEMPONE.NDX
59 SET CONSOLE ON
60 USE SERIALNO INDEX SERNSIT
61 *
62 DO WHILE .T.
63     SET COLOR TO /BR, /BR
64     STORE LOSITE TO MSITE
65     @ 13,66 GET MSITE PICT '99'
66     READ
67     IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
68         SET COLOR TO W+/R, W+/R
69         STORE ' Response must be between ' + LOSITE + ;
70             ' and ' + HISITE + ' ' TO ERROR
71         @ 24,22 SAY ERROR
72         DO DELAY
73         LOOP
74     ELSE
75         GO TOP
76         FIND &MSITE
77         IF EOF() = .T. THEN
78             STORE " No ser al numbers exist for site " + MSITE + ;
79                 ", try another site " TO MESSAGE
80             SET COLOR TO W+/R, W+/R
81             @ 24,13 SAY MESSAGE
82             DO DELAY
83             LOOP
84         ELSE
85             EXIT
86         ENDIF EOF() = .T.
87     ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
88 ENDDO WHILE .T.
89 *
90 SET COLOR TO W+/BR, W+/BR
91 @ 13,15 SAY SPACE(60)
92 *
93 SET COLOR TO W+/B, W+/B
94 @ 05,09 SAY "The following Delivery Order Effective Dates exist for Site"
95 @ 05,69 SAY MSITE
96 SET COLOR TO /BR, /BR
97 @ 13,05 SAY SPACE(70)
98 STORE 1 TO LINECT
99 STORE 1.00 TO COLCNT
100 STORE "000000" TO MOLDATE

```

Page 3

SNODTRPT.PRG Program Listing

```
101 *
102 DO WHILE SITENO = MSITE
103   IF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00) THEN
104     @LINECT+6,57 SAY EFFDATE
105   ELSE
106     IF (COLCNT - (COLCNT * (COLCNT/2)) = 0.00) THEN
107       @LINECT+6,38 SAY EFFDATE
108     ELSE
109       @LINECT+6,19 SAY EFFDATE
110     ENDIF (COLCNT - (COLCNT * (COLCNT/2)) = 0.00)
111   ENDIF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00)
112   IF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00) THEN
113     LINECT = 1 + LINECT
114     COLCNT = 1.00
115   ELSE
116     COLCNT = COLCNT + 1.00
117   ENDIF (COLCNT - (COLCNT * (COLCNT/3)) = 0.00)
118   STORE EFFDATE TO MOLDATE
119 *
120   DO WHILE ((EFFDATE = MOLDATE) .AND. .NOT. EOF())
121     SKIP+2
122   ENDDO
123 *
124   IF EOF() THEN
125     EXIT
126   ELSE
127     SKIP
128   ENDIF EOF() = .T.
129   ENDDO WHILE SITENO = MSITE
130 *
131   STORE DTOC(DATE()) TO SYSDATE
132   STORE SUBSTR(SYSDATE,7,2) + SUBSTR(SYSDATE,1,2) + ;
133     SUBSTR(SYSDATE,4,2) TO MDATE
134   STORE SPACE(17) + 'Input Effective Date (Range ' + LODATE + ;
135     ' to ' + HIDATE + ')' + SPACE(17) TO MESSAGE
136   SET COLOR TO /W, /W
137   @ 24,0 SAY MESSAGE
138   SET COLOR TO W+/B, W+/B
139   @ 3,29 SAY "EFFECTIVE DATE: "
140 *
141   USE SERIALNO INDEX SERNODAT
142   STORE "000000" TO MOLDATE
143 *
144   DO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDATE)
145     STORE MDATE TO MOLDATE
146     SET COLOR TO R+/B, R+/B
147     @ 3,45 GET MOLDATE PICT "999999"
148     READ
149     LO WHILE .T.
150       IF .NOT. (SUBSTR(MOLDATE,1,2) > "83" .AND.;
```

```

151          SUBSTR(MOLDATE,1,2) <= "99") THEN
152          SET COLOR TO W/B, W/B
153          @ 24,0 SAY SPACE(80)
154          SET COLOR TO W+/R, W+/R
155          @ 24,16 SAY " Year portion of date must be between 84 and 99 "
156          DO DELAY
157          SET COLOR TO /W, /W
158          @ 24,0 SAY MESSAGE
159          STORE SUBSTR(MDATE,1,2) TO MYEAR
160          SET COLOR TO R+/B, R+/B
161          @ 3,45 GET MYEAR PICT "99"
162          READ
163          STORE MYEAR + SUBSTR(MOLDATE,3,4) TO MOLDATE
164      ELSE
165          EXIT
166      ENDIF
167  ENDDO WHILE .T.
168  *
169  DO WHILE .T.
170      IF .NOT. (SUBSTR(MOLDATE,3,2) >= "01" .AND.;
171              SUBSTR(MOLDATE,3,2) <= "12") THEN
172          SET COLOR TO W/B, W/B
173          @ 24,0 SAY SPACE(80)
174          SET COLOR TO W+/R, W+/R
175          @ 24,16 SAY " Month portion of date must be between 01 and 12 "
176          DO DELAY
177          SET COLOR TO /W, /W
178          @ 24,0 SAY MESSAGE
179          STORE SUBSTR(MDATE,3,2) TO MMONTH
180          SET COLOR TO R+/B, R+/B
181          @ 3,47 GET MMONTH PICT "99"
182          READ
183          STORE SUBSTR(MOLDATE,1,2) + MMONTH +;
184              SUBSTR(MOLDATE,5,2) TO MOLDATE
185      ELSE
186          EXIT
187      ENDIF
188  ENDDO WHILE .T.
189  *
190  DO WHILE .T.
191      IF ((SUBSTR(MOLDATE,3,2) = "04" .OR. SUBSTR(MOLDATE,3,2) = "06" .OR.;
192          SUBSTR(MOLDATE,3,2) = "09" .OR. SUBSTR(MOLDATE,3,2) = "11") .AND. .NOT.;
193          (SUBSTR(MOLDATE,5,2) >= "01" .AND. SUBSTR(MOLDATE,5,2) <= "30")) THEN
194          SET COLOR TO W/B, W/B
195          @ 24,0 SAY SPACE(80)
196          SET COLOR TO W+/R, W+/R
197          @ 24,16 SAY " Day portion of date must be between 01 and 30 "
198          DO DELAY
199          SET COLOR TO /W, /W
200          @ 24,0 SAY MESSAGE

```

```
201     STORE SUBSTR(MDATE,5,2) TO MDAY
202     SET COLOR TO R+/B, R+B
203     @ 3,49 GET MDAY PICT "99"
204     READ
205     STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
206     LOOP
207     ELSE
208     *
209     IF (SUBSTR(MOLDATE,3,2) = "02" .AND. .NOT.;
210        (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
211        SUBSTR(MOLDATE,5,2) <= "28")) THEN
212        SET COLOR TO W/B, W/B
213        @ 24,0 SAY SPACE(80)
214        SET COLOR TO W+/R, W+/R
215        @ 24,16 SAY " Day portion of date must be between 01 and 28 "
216        DO DELAY
217        SET COLOR TO /W, /W
218        @ 24,0 SAY MESSAGE
219        STORE SUBSTR(MDATE,5,2) TO MDAY
220        SET COLOR TO R+/B, R+B
221        @ 3,49 GET MDAY PICT "99"
222        READ
223        STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
224        LOOP
225        ELSE
226        *
227        IF .NOT. (SUBSTR(MOLDATE,5,2) >= "01" .AND.;
228                SUBSTR(MOLDATE,5,2) <= "31") THEN
229                SET COLOR TO W/B, W/B
230                @ 24,0 SAY SPACE(80)
231                SET COLOR TO W+/R, W+/R
232                @ 24,16 SAY " Day portion of date must be between 01 and 31 "
233                DO DELAY
234                SET COLOR TO /W, /W
235                @ 24,0 SAY MESSAGE
236                STORE SUBSTR(MDATE,5,2) TO MDAY
237                SET COLOR TO R+/B, R+B
238                @ 3,49 GET MDAY PICT "99"
239                READ
240                STORE SUBSTR(MOLDATE,1,4) + MDAY TO MOLDATE
241                LOOP
242        ELSE
243        EXIT
244        ENDIF
245        ENDIF
246        ENDIF
247        ENDDO WHILE .T.
248    *
249    GO TOP
250    STORE MSITE + MOLDATE TO MKEY
```

```
251 FIND &MKEY
252 IF EOF() = .T. THEN
253     SET COLOR TO W/B, W/B
254     @ 24,0 SAY SPACE(80)
255     STORE "EFFECTIVE DATE " + MOLDATE + " does not exist for site " +
256         MSITE + ", try another " TO NODATE
257     SET COLOR TO W+/R, W+/R
258     @ 24,10 SAY NODATE
259     DO DELAY
260     SET COLOR TO /W, /W
261     @ 24,0 SAY MESSAGE
262     STORE "000000" TO MOLDATE
263     LOOP
264 ELSE
265     EXIT
266 ENDIF EOF() = .T.
267 ENDDO WHILE .NOT. (MOLDATE >= LODATE .AND. MOLDATE <= HIDATE)
268 *
269 SET COLOR TO W+/B, W+/B
270 @ 05,05 SAY SPACE(70)
271 @ 24,0 SAY SPACE(80)
272 *
273 * CLEAR LISTING OF EFFECTIVE DATES FROM SCREEN
274 *
275 SET COLOR TO /BR, /BR
276 @ 07,2 SAY SPACE(76)
277 @ 08,2 SAY SPACE(76)
278 @ 09,2 SAY SPACE(76)
279 @ 10,2 SAY SPACE(76)
280 @ 11,2 SAY SPACE(76)
281 @ 12,2 SAY SPACE(76)
282 @ 13,2 SAY SPACE(76)
283 @ 14,2 SAY SPACE(76)
284 @ 15,2 SAY SPACE(76)
285 @ 16,2 SAY SPACE(76)
286 @ 17,2 SAY SPACE(76)
287 @ 18,2 SAY SPACE(76)
288 @ 19,2 SAY SPACE(76)
289 @ 20,2 SAY SPACE(76)
290 @ 21,2 SAY SPACE(76)
291 *
292 SET COLOR TO R+/ , R+/
293 @ 13,18 SAY " CREATING TEMPORARY DATABASE AND INDEX FILE "
294 *
295 COPY TO TEMPONE FOR SITENO = "&MSITE" .AND. EFFDATE = "&MOLDATE"
296 SELECT 1
297 USE TEMPONE
298 INDEX ON FEATURENO TO TEMPONE
299 SELECT 2
300 USE DESCRIP INDEX DESCRIP
```

```

301 SELECT TEMPONE
302 SET RELATION TO FEATURENO INTO DESCRIP
303 GO TOP
304 *
305 *   CREATE THE SPLICE EQUIPMENT PROJECT REPORT AND CHECK IF THE REPORT
306 *   IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
307 *
308 SET COLOR TO W+/BR, W+/BR
309 @ 13,15 SAY SPACE(60)
310 @ 13,16 SAY " Do you want a printed report? (Yes or No): "
311 SET COLOR TO /BR, /BR
312 @ 13,49 SAY "Y"
313 @ 13,56 SAY "N"
314 STORE "N" TO ACCEPT
315 @ 13,62 GET ACCEPT PICT "!"
316 READ
317 *
318 *   ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
319 *
320 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
321   IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
322     SET COLOR TO W+/R, W+/R
323     @ 24,24 SAY " Response must be either N or Y "
324     DO DELAY
325     STORE "N" TO ACCEPT
326   ENDIF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
327   SET COLOR TO /BR, /BR
328   @ 13,62 GET ACCEPT PICT "!"
329   READ
330 ENDDO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
331 *
332 SET COLOR TO /BR, /BR
333 @ 13,15 SAY SPACE(55)
334 *
335 IF ACCEPT = "Y" THEN
336   ?? FLASH + "W.PRINTER/"
337   SET CONSOLE OFF
338   WAIT TO CHOICE
339   SET CONSOLE ON
340   SET COLOR TO W/B, W/B
341   @ 22,10 SAY SPACE(65)
342   STORE DUC( DATE() ) TO TODAY
343   STORE SUBSTR(TODAY,4,2) + " " + MONTH( DATE() ) + " 19" +
344     SUBSTR(TODAY,7,2) TO TODAY
345   STORE 0 TO PAGENO
346   STORE 0 TO LINECT
347   SET COLOR TO R+/ , R+/
348   SET DEVICE TO PRINT
349 *
350 DO WHILE .NOT. EOF()

```

```

351 DO WHILE (LINECT <= 60 .AND. .NOT. EOF())
352     @ LINECT,3 SAY SITENO PICT "99"
353     @ LINECT,7 SAY B->CLIN PICT "9999"
354     @ LINECT,15 SAY FEATURENO PICT "999999"
355     @ LINECT,24 SAY B->DESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
356     @ LINECT,52 SAY EFFDATE PICT "999999"
357     @ LINECT,60 SAY TOTQTY PICT "999"
358     @ LINECT,65 SAY QTY PICT "999"
359     @ LINECT,70 SAY SERIALNO PICT "!!!!!!!!!!"
360     LINECT = LINECT + 1
361     SKIP
362 ENDDO WHILE (LINECT <= 60 .AND. .NOT. EOF())
363 *
364 IF EOF() = .T. THEN
365     IF PAGENO > 1 THEN
366         @ 62,37 SAY "Page " + STR(PAGENO,2,0)
367     ENDIF PAGENO > 1
368     EJECT
369     SET DEVICE TO SCREEN
370     @ 13,25 SAY " FINISHED PRINTING THE REPORT "
371     DO DELAY
372     EXIT
373 ELSE
374     SET DEVICE TO SCREEN
375     @ 13,27 SAY " Printing Page Number " + STR(PAGENO + 1,2,0) + " "
376     SET DEVICE TO PRINT
377 ENDIF EOF() = .T.
378 *
379 IF (LINECT > 60 .AND. PAGENO > 1) THEN
380     @ 62,37 SAY "Page " + STR(PAGENO,2,0)
381 ENDIF (LINECT > 60 .AND. PAGENO > 1)
382 @ 2,26 SAY " SITE SERIAL NUMBER REPORT "
383 @ 3,29 SAY "EFFECTIVE DATE: "
384 @ 3,45 SAY MOLDATE
385 @ 4,60 SAY TODATE
386 @ 6,52 SAY "EFFECT TOT COMPT SERIAL"
387 @ 7,2 SAY "SITE CLIN FEATURE# DESCRIPTION DATE"
388 @ 7,60 SAY "QTY QTY NUMBER"
389 @ 8,2 SAY "===== "
390 @ 8,51 SAY "===== "
391 PAGENO = PAGENO + 1
392 STORE 10 TO LINECT
393 *
394 ENDDO WHILE .NOT. EOF()
395 ELSE
396 SET COLOR TO GR+/B, GR+/B
397 @ 4,52 SAY "EFFECT TOT COMPT SERIAL"
398 @ 5,2 SAY "SITE CLIN FEATURE# DESCRIPTION DATE"
399 @ 5,60 SAY "QTY QTY NUMBER"
400 SET COLOR TO /BR, /BR

```



```

401 STORE 0 TO LINECT
402 *
403 DO WHILE .NOT. EOF()
404     DO WHILE LINECT < 15
405         @ LINECT+7,3 SAY SITENO PICT "99"
406         @ LINECT+7,7 SAY B->CLIN PICT "9999"
407         @ LINECT+7,15 SAY FEATURENO PICT "999999"
408         @ LINECT+7,24 SAY B->DESCRIPT PICT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
409         @ LINECT+7,52 SAY EFFDATE PICT "999999"
410         @ LINECT+7,60 SAY TOTQTY PICT "999"
411         @ LINECT+7,65 SAY QTY PICT "999"
412         @ LINECT+7,70 SAY SERIALNO PICT "!!!!!!!!!!"
413         LINECT = LINECT + 1
414         SKIP
415         IF EOF() = .T. THEN
416             SET COLOR TO W+/R, W+/R
417             @ 24,18 SAY " End of File reached, Press any key to EXIT "
418             SET CONSOLE OFF
419             WAIT TO ACCEPT
420             SET CONSOLE ON
421             EXIT
422         ENDIF EOF() = .T.
423     ENDDO WHILE LINECT < 15
424 *
425 IF EOF() = .T. THEN
426     EXIT
427 ENDIF EOF() = .T.
428 SET COLOR TO R+/B, R+/B
429 STORE "C" TO CHOICE
430 @ 22,57 GET CHOICE PICT "!"
431 READ
432 *
433 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
434 *
435 DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
436     IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
437         SET COLOR TO W+/R, W+/R
438         @ 24,24 SAY " Response must be either C or X "
439         DO DELAY
440         STORE "C" TO CHOICE
441     ENDIF .NOT. (CHOICE = "C" .OR. CHOICE = "X")
442     SET COLOR TO R+/B, R+/B
443     @ 22,57 GET CHOICE PICT "!"
444     READ
445 ENDDO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
446 *
447 * DETERMINE IF THE USER WANTS TO QUIT OR CONTINUE
448 *
449 IF CHOICE = "C"
450     SET COLOR TO /BR, /BR

```

Page 10

SNODTRPT.PRG Program Listing

```
451      @ 07,2 SAY SPACE(76)
452      @ 08,2 SAY SPACE(76)
453      @ 09,2 SAY SPACE(76)
454      @ 10,2 SAY SPACE(76)
455      @ 11,2 SAY SPACE(76)
456      @ 12,2 SAY SPACE(76)
457      @ 13,2 SAY SPACE(76)
458      @ 14,2 SAY SPACE(76)
459      @ 15,2 SAY SPACE(76)
460      @ 16,2 SAY SPACE(76)
461      @ 17,2 SAY SPACE(76)
462      @ 18,2 SAY SPACE(76)
463      @ 19,2 SAY SPACE(76)
464      @ 20,2 SAY SPACE(76)
465      @ 21,2 SAY SPACE(76)
466      STORE 0 TO LINECT
467      ELSE .
468          EXIT
469      ENDIF CHOICE = "C"
470      *
471      ENDDO WHILE .NOT. EOF()
472      *
473      ENDIF ACCEPT = "Y"
474      *
475      * ERASE ALL TEMPORARY FILES CREATED DURING REPORT GENERATION
476      *
477      CLOSE DATABASES
478      SET CONSOLE OFF
479      ERASE TEMPONE.DBF
480      ERASE TEMPONE.NDX
481      SET CONSOLE ON
482      SET PRINT OFF
483      *
484      * RETURN TO CALLING PROGRAM
485      *
486      RELEASE ALL LIKE M*, ACCEPT, CHOICE, COLCNT, LINECT, PAGENO,;
487      SYSDATE, TODAY, TODATE
488      RETURN
489      *****
```

```

1  * PROCEDURE SNOPJRPT.PRG
2  *
3  * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4  *              LCDR WINSTON H. BUCKLEY, SC, USN
5  *              LCDR ROBERT F. BRADO, USN
6  *              LCDR ROBERT L. BEARD III, SC, USN
7  *
8  * PURPOSE      : PROVIDE THE USER A SPLICE SERIAL NUMBER
9  *              PROJECT LEVEL REPORT.
10 *
11 * INPUT FILES   : SERIALNO.DBF, SERNOPRJ.NDX, DESCRIP.DBF, DESCRIP.NDX
12 *
13 * OUTPUT FILES  : NONE.
14 *
15 * CALLED BY     : PROJRPPTS.PRG
16 *
17 * MODULES CALLED : DELAY.PRG
18 *
19 * LOCAL VARIABLES: ACCEPT, CHOICE, LINECT, PAGENO, TODAY, TODATE
20 *
21 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
22 *
23 * CASE SELECTION = 2      SERIAL NUMBER PROJECT LEVEL REPORT
24 *
25 * CALL SERIAL NUMBER DATABASE INDEXED ON EFFECTIVE DATE, SITE NUMBER,
26 * AND FEATURE NUMBER.  RELATE TO DESCRIP FILE ON FEATURENO.
27 *
28 SET ESCAPE OFF
29 SET TALK OFF
30 SET COLOR TO W+/B, W+/B, B
31 CLEAR
32 USE SERIALNO
33 GO TOP
34 IF EOF() = .T. THEN
35     SET COLOR TO W+/R, W+/R
36     @ 13,22 SAY " The SERIAL NUMBER Database is EMPTY! "
37     DO DELAY
38     RETURN
39 ENDIF
40 ?? FLASH + "S.REPORTS.SCR/"
41 @ 24,0 SAY SPACE(80)
42 SET COLOR TO R+/ , R+/
43 @ 2,18 SAY " EQUIPMENT SERIAL NUMBER PROJECT LEVEL REPORT "
44 SELECT 1
45 USE SERIALNO INDEX SERNOPRJ.NDX
46 SELECT 2
47 USE DESCRIP INDEX DESCRIP
48 SELECT SERIALNO
49 SET RELATION 4 TO FEATURENO INTO DESCRIP
50 * END

```

```

51 *
52 * CREATE THE SPLICE SERIAL NUMBER PROJECT REPORT AND CHECK IF THE REPORT
53 * IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
54 *
55 SET COLOR TO W+/BR, W+/BR
56 @ 13,16 SAY " Do you want a printed report? (Yes or No): "
57 SET COLOR TO /BR, /BR
58 @ 13,49 SAY "Y"
59 @ 13,56 SAY "N"
60 STORE "N" TO ACCEPT
61 @ 13,62 GET ACCEPT PICT "!"
62 READ
63 *
64 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
65 *
66 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
67   IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
68     SET COLOR TO W+/R, W+/R
69     @ 24,24 SAY " Response must be either N or Y "
70     DO DELAY
71     STORE "N" TO ACCEPT
72   ENDIF
73   SET COLOR TO /BR, /BR
74   @ 13,62 GET ACCEPT PICT "!"
75   READ
76 ENDDO
77 *
78 SET COLOR TO /BR, /BR
79 @ 13,15 SAY SPACE(55)
80 *
81 IF ACCEPT = "Y" THEN
82   ?? FLASH + "W.PRINTER/"
83   SET CONSOLE OFF
84   WAIT TO CHOICE
85   SET CONSOLE ON
86   SET COLOR TO W/B, W/B
87   @ 22,10 SAY SPACE(65)
88   STORE 0 TO PAGENO
89   STORE 61 TO LINECT
90   STORE DTOC( DATE() ) TO TODAY
91   STORE SUBSTR(TODAY,4,2) + " " + CMONTH( DATE() ) + " 19" + ;
92     SUBSTR(TODAY,7,2) TO TDATE
93   SET COLOR TO R+/ , R+/
94   SET DEVICE TO PRINT
95 *
96 DO WHILE .NOT. EOF()
97   DO WHILE (LINECT <= 60 .AND. .NOT. EOF())
98     @ LINECT,3 SAY SITENO
99     @ LINECT,7 SAY DESCRIP->CLIN
100    @ LINECT,15 SAY FEATURENO

```

```
101      @ LINECT,24 SAY DESCRIP->DESCRIPT
102      @ LINECT,52 SAY EFFDATE
103      @ LINECT,60 SAY TOTQTY
104      @ LINECT,65 SAY QTY
105      @ LINECT,70 SAY SERIALNO
106      LINECT = LINECT + 1
107      SKIP
108      ENDDO WHILE
109  *
110      IF EOF() = .T. THEN
111          IF PAGENO > 1 THEN
112              @ 62,37 SAY "Page " + STR(PAGENO,2,0)
113          ENDIF
114          EJECT
115          SET DEVICE TO SCREEN
116          @ 13,25 SAY " FINISHED PRINTING THE REPORT "
117          DO DELAY
118          EXIT
119      ELSE
120          SET DEVICE TO SCREEN
121          @ 13,27 SAY " Printing Page Number " + STR(PAGENO + 1,2,0) + " "
122          SET DEVICE TO PRINT
123      ENDIF
124  *
125      IF (LINECT > 60 .AND. PAGENO > 1) THEN
126          @ 62,37 SAY "Page " + STR(PAGENO,2,0)
127      ENDIF
128      @ 2,18 SAY " EQUIPMENT SERIAL NUMBER PROJECT LEVEL REPORT "
129      @ 4,62 SAY TODATE
130      @ 6,52 SAY "EFFECT TOT COMPT SERIAL"
131      @ 7,2 SAY "SITE CLIN FEATURE#          DESCRIPTION          DATE"
132      @ 7,60 SAY "QTY QTY NUMBER"
133      @ 8,2 SAY "===== "
134      @ 8,51 SAY "===== "
135      PAGENO = PAGENO + 1
136      STORE 10 TO LINECT
137  *
138      ENDDO WHILE .NOT. EOF()
139  *
140      ELSE
141          SET COLOR TO GR+/B, GR+/B
142          @ 4,52 SAY "EFFECT TOT COMPT SERIAL"
143          @ 5,2 SAY "SITE CLIN FEATURE#          DESCRIPTION          DATE"
144          @ 5,60 SAY "QTY QTY NUMBER"
145          SET COLOR TO /BR, /BR
146          STORE 0 TO LINECT
147  *
148          DO WHILE .NOT. EOF()
149              DO WHILE LINECT < 15
150                  @ LINECT+7,3 SAY SITENO
```

Page 4

SNOPJRPT.PRG Program Listing

```
151 @ LINECT+7,7 SAY DESCRIP->CLIN
152 @ LINECT+7,15 SAY FEATURENO
153 @ LINECT+7,24 SAY DESCRIP->DESCRIPT
154 @ LINECT+7,52 SAY EFFDATE
155 @ LINECT+7,60 SAY TOTQTY
156 @ LINECT+7,65 SAY QTY
157 @ LINECT+7,70 SAY SERIALNO
158 LINECT = LINECT + 1
159 SKIP
160 IF EOF() = .T. THEN
161     SET COLOR TO W+/R, W+/R
162     @ 24,18 SAY " End of File reached, Press any key to EXIT "
163     SET CONSOLE OFF
164     WAIT TO ACCEPT
165     SET CONSOLE ON
166     EXIT
167 ENDIF
168 ENDDO WHILE LINECT < 15
169 *
170 IF EOF() = .T. THEN
171     EXIT
172 ENDIF
173 SET COLOR TO R+/B, R+/B
174 STORE "C" TO CHOICE
175 @ 22,57 GET CHOICE PICT "!"
176 READ
177 *
178 * ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
179 *
180 DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
181     IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
182         SET COLOR TO W+/R, W+/R
183         @ 24,24 SAY " Response must be either C or X "
184         DO DELAY
185         STORE "C" TO CHOICE
186     ENDIF
187     SET COLOR TO R+/B, R+/B
188     @ 22,57 GET CHOICE PICT "!"
189     READ
190 ENDDO
191 *
192 * DETERMINE IF THE USER WANTS TO QUIT OR CONTINUE
193 *
194 IF CHOICE = "C"
195     SET COLOR TO /BR, /BR
196     @ 07,2 SAY SPACE(76)
197     @ 08,2 SAY SPACE(76)
198     @ 09,2 SAY SPACE(76)
199     @ 10,2 SAY SPACE(76)
200     @ 11,2 SAY SPACE(76)
```

```
201      @ 12,2 SAY SPACE(76)
202      @ 13,2 SAY SPACE(76)
203      @ 14,2 SAY SPACE(76)
204      @ 15,2 SAY SPACE(76)
205      @ 16,2 SAY SPACE(76)
206      @ 17,2 SAY SPACE(76)
207      @ 18,2 SAY SPACE(76)
208      @ 19,2 SAY SPACE(76)
209      @ 20,2 SAY SPACE(76)
210      @ 21,2 SAY SPACE(76)
211      STORE 0 TO LINECT
212      ELSE
213          EXIT
214      ENDIF
215      *
216      ENDDO WHILE .NOT. EOF()
217      *
218      ENDIF
219      *
220      * RETURN TO CALLING PROGRAM
221      *
222      SET PRINT OFF
223      RELEASE ACCEPT, CHOICE, LINECT, PAGENO, TODAY, TODATE
224      CLOSE DATABASES
225      RETURN
226      *****
```

Page 1

SNOSTRPT.PRG Program Listing

```
1 * PROCEDURE SNOSTRPT.PRG
2 *
3 * AUTHORS      : LCDR EDWARD J. CASE, SC, USN
4 *              LCDR WINSTON H. BUCKLEY, SC, USN
5 *              LCDR ROBERT F. BRADO, USN
6 *              LCDR ROBERT L. BEARD III, SC, USN
7 *
8 * PURPOSE      : PROVIDE THE USER A SPLICE SERIAL NUMBER
9 *              SITE LEVEL REPORT.
10 *
11 * INPUT FILES  : SERIALNO.DBF, SERNOSIT.NDX, DESCRIP.DBF,
12 *              DESCRIP.NDX
13 *
14 * CALLED BY    : SITERPTS.PRG
15 *
16 * MODULES CALLED : DELAY.PRG
17 *
18 * GLOBAL VARIABLE: HISITE, LOSITE
19 *
20 * LOCAL VARIABLES: ACCEPT, CHOICE, ERROR, LINECT, MESSAGE, MSITE,
21 *                 PAGENO, TODAY, TODATE
22 *
23 * DATE LAST TIME MODIFIED =====> 27 DECEMBER 1985 <=====
24 *
25 * CASE SELECTION = 3      SERIAL NUMBER SITE LEVEL REPORT
26 *
27 SET ESCAPE OFF
28 SET TALK OFF
29 SET COLOR TO W+/B, W+/B, B
30 CLEAR
31 USE SERIALNO
32 GO TOP
33 IF EOF() = .T. THEN
34     SET COLOR TO W+/R, W+/R
35     @ 13,22 SAY " The SERIAL NUMBER Database is EMPTY! "
36     DO DELAY
37     RETURN
38 ENDIF
39 ?? FLASH + "S.REPORTS.SCR/"
40 @ 24,0 SAY SPACE(80)
41 SET COLOR TO R+/ , R+/
42 @ 2,26 SAY " SITE SERIAL NUMBER REPORT "
43 SET COLOR TO W+/BR, W+/BR
44 @ 13,15 SAY "Enter site number for which the report is desired:"
45 *
46 * CALL SERIAL NUMBER DATABASE INDEXED ON SITE NUMBER,
47 * FEATURE NUMBER AND SERIAL NUMBER.  RELATE TO DESCRIPTION FILE.
48 *
49 SELECT 1
50 USE SERIALNO INDEX SERNOSIT.NDX
```


Page 2

SNOSTRPT.PRG Program Listing

```
51 SELECT 2
52 USE DESCRIP INDEX DESCRIP
53 SELECT SERIALNO
54 SET RELATION TO FEATURENO INTO DESCRIP
55 *
56 DO WHILE .T.
57     SET COLOR TO /BR, /BR
58     STORE LOSITE TO MSITE
59     @ 13,66 GET MSITE PICT '99'
60     READ
61     IF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE) THEN
62         SET COLOR TO W+/R, W+/R
63         STORE ' Response must be between ' + LOSITE + ;
64             ' and ' + HISITE + ' ' TO ERROR
65         @ 24,22 SAY ERROR
66         DO DELAY
67         LOOP
68     ELSE
69         GO TOP
70         FIND &MSITE
71         IF EOF() = .T. THEN
72             STORE " No serial numbers exist for site " + MSITE + ;
73                 ", try another site " TO MESSAGE
74             SET COLOR TO W+/R, W+/R
75             @ 24,13 SAY MESSAGE
76             DO DELAY
77             LOOP
78         ELSE
79             EXIT
80         ENDIF EOF() = .T.
81     ENDIF .NOT. (MSITE >= LOSITE .AND. MSITE <= HISITE)
82 ENDDO WHILE .T.
83 *
84 SET COLOR TO W+/BR, W+/BR
85 @ 13,15 SAY SPACE(60)
86 *
87 *   CREATE THE SPLICE EQUIPMENT PROJECT REPORT AND CHECK IF THE REPORT
88 *   IS TO BE PRINTED OR DISPLAYED ON THE SCREEN.
89 *
90 @ 13,16 SAY " Do you want a printed report? (Yes or No): "
91 SET COLOR TO /BR, /BR
92 @ 13,49 SAY "Y"
93 @ 13,56 SAY "N"
94 STORE "N" TO ACCEPT
95 @ 13,62 GET ACCEPT PICT "!"
96 READ
97 *
98 *   ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
99 *
100 DO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
```

```

101 IF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y") THEN
102     SET COLOR TO W+/R, W+/R
103     @ 24,24 SAY " Response must be either N or Y "
104     DO DELAY
105     STORE "N" TO ACCEPT
106 ENDIF .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
107 SET COLOR TO /BR, /BR
108 @ 13,62 GET ACCEPT PICT "!"
109 READ
110 ENDDO WHILE .NOT. (ACCEPT = "N" .OR. ACCEPT = "Y")
111 *
112 SET COLOR TO /BR, /BR
113 @ 13,15 SAY SPACE(55)
114 *
115 IF ACCEPT = "Y" THEN
116     ?? FLASH + "W.PRINTER/"
117     SET CONSOLE OFF
118     WAIT TO CHOICE
119     SET CONSOLE ON
120     SET COLOR TO W/B, W/B
121     @ 22,10 SAY SPACE(65)
122     STORE DIOC( DATE() ) TO TODAY
123     STORE SUBSTR(TODAY,4,2) + " " + CMONTH( DATE() ) + " 19" +;
124         SUBSTR(TODAY,7,2) TO TODATE
125     STORE 0 TO PAGENO
126     STORE 61 TO LINECT
127     SET COLOR TO R+/ , R+/
128     SET DEVICE TO PRINT
129 *
130 DO WHILE .NOT. EOF()
131     DO WHILE (LINECT <= 60 .AND. .NOT. EOF())
132         @ LINECT,3 SAY SITENO
133         @ LINECT,7 SAY DESCRIP->CLIN
134         @ LINECT,15 SAY FEATURENO
135         @ LINECT,24 SAY DESCRIP->DESCRIP
136         @ LINECT,52 SAY EFFDATE
137         @ LINECT,60 SAY TOTQTY
138         @ LINECT,65 SAY QTY
139         @ LINECT,70 SAY SERIALNO
140         LINECT = LINECT + 1
141     SKIP
142     ENDDO WHILE WHILE (LINECT <= 60 .AND. .NOT. EOF())
143 *
144 IF EOF() = .T. THEN
145     IF PAGENO > 1 THEN
146         @ 62,37 SAY "Page " + STR(PAGENO,2,0)
147     ENDIF PAGENO > 1
148     EJECT
149     SET DEVICE TO SCREEN
150     @ 13,25 SAY " FINISHED PRINTING THE REPORT "

```

```

151      DO DELAY
152      EXIT
153  ELSE
154      SET DEVICE TO SCREEN
155      @ 13,27 SAY " Printing Page Number " + STR(PAGENO + 1,2,0) + " "
156      SET DEVICE TO PRINT
157  ENDIF EOF() = .T.
158  *
159      IF (LINECT > 60 .AND. PAGENO > 1) THEN
160          @ 62,37 SAY "Page " + STR(PAGENO,2,0)
161      ENDIF (LINECT > 60 .AND. PAGENO > 1)
162      @ 2,26 SAY " SITE SERIAL NUMBER REPORT "
163      @ 4,60 SAY TODATE
164      @ 6,52 SAY "EFFECT TOT COMPT SERIAL"
165      @ 7,2 SAY "SITE CLIN  FEATURE#          DESCRIPTION          DATE"
166      @ 7,60 SAY "QTY QTY  NUMBER"
167      @ 8,2 SAY "===== "
168      @ 8,51 SAY "===== "
169      PAGENO = PAGENO + 1
170      STORE 10 TO LINECT
171  *
172  ENDDO WHILE .NOT. EOF()
173  *
174  ELSE
175      SET COLOR TO GR+/B, GR+/B
176      @ 4,52 SAY "EFFECT TOT COMPT SERIAL"
177      @ 5,2 SAY "SITE CLIN  FEATURE#          DESCRIPTION          DATE"
178      @ 5,60 SAY "QTY QTY  NUMBER"
179      SET COLOR TO /BR, /BR
180      STORE 0 TO LINECT
181  *
182  DO WHILE .NOT. EOF()
183      DO WHILE LINECT < 15
184          @ LINECT+7,3 SAY SITENO
185          @ LINECT+7,7 SAY DESCRIP->CLIN
186          @ LINECT+7,15 SAY FEATURENO
187          @ LINECT+7,24 SAY DESCRIP->DESCRIPT
188          @ LINECT+7,52 SAY EFFDATE
189          @ LINECT+7,60 SAY TOTQTY
190          @ LINECT+7,65 SAY QTY
191          @ LINECT+7,70 SAY SERIALNO
192          LINECT = LINECT + 1
193          SKIP
194      IF EOF() = .T. THEN
195          SET COLOR TO W+/R, W+/R
196          @ 24,18 SAY " End of File reached, Press any key to EXIT "
197          SET CONSOLE OFF
198          WAIT TO ACCEPT
199          SET CONSOLE ON
200          EXIT

```

Page 5

SNOSTRPT.PRG Program Listing

```
201         ENDIF EOF() = .T.
202     ENDDO WHILE LINECT < 15
203 *
204     IF EOF() = .T. THEN
205         EXIT
206     ENDIF EOF() = .T.
207     SET COLOR TO R+/B, R+/B
208     STORE "C" TO CHOICE
209     @ 22,57 GET CHOICE PICT "!"
210     READ
211 *
212 *     ENSURE THAT THE USER'S RESPONSE IS EITHER "Y" OR "N"
213 *
214     DO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
215         IF .NOT. (CHOICE = "C" .OR. CHOICE = "X") THEN
216             SET COLOR TO W+/R, W+/R
217             @ 24,24 SAY " Response must be either C or X "
218             DO DELAY
219             STORE "C" TO CHOICE
220         ENDIF .NOT. (CHOICE = "C" .OR. CHOICE = "X")
221         SET COLOR TO R+/B, R+/B
222         @ 22,57 GET CHOICE PICT "!"
223         READ
224     ENDDO WHILE .NOT. (CHOICE = "C" .OR. CHOICE = "X")
225 *
226 *     DETERMINE IF THE USER WANTS TO QUIT OR CONTINUE
227 *
228     IF CHOICE = "C"
229         SET COLOR TO /BR, /BR
230         @ 07,2 SAY SPACE(76)
231         @ 08,2 SAY SPACE(76)
232         @ 09,2 SAY SPACE(76)
233         @ 10,2 SAY SPACE(76)
234         @ 11,2 SAY SPACE(76)
235         @ 12,2 SAY SPACE(76)
236         @ 13,2 SAY SPACE(76)
237         @ 14,2 SAY SPACE(76)
238         @ 15,2 SAY SPACE(76)
239         @ 16,2 SAY SPACE(76)
240         @ 17,2 SAY SPACE(76)
241         @ 18,2 SAY SPACE(76)
242         @ 19,2 SAY SPACE(76)
243         @ 20,2 SAY SPACE(76)
244         @ 21,2 SAY SPACE(76)
245         STORE 0 TO LINECT
246     ELSE
247         EXIT
248     ENDIF CHOICE = "C"
249 *
250     ENDDO WHILE .NOT. EOF()
```

Page 6

SNOSTRPT.PRG Program Listing

```
251  *
252  ENDIF ACCEPT = "Y"
253  *
254  * RETURN TO CALLING PROGRAM
255  *
256  SET PRINT OFF
257  RELEASE ACCEPT, CHOICE, ERROR, LINECT, MESSAGE, MSITE, PAGENO,;
258      TODAY, TODATE
259  CLOSE DATABASES
260  RETURN
261  *****
262
```

BIBLIOGRAPHY

Alter, Steven, Decision Support Systems, Addison-Wesley, New York, 1980.

Alter, Steven, "A Taxonomy of Decision Support Systems," Sloan Management Review, Vol. 19, No. 1, Fall 1981, pp. 39-56.

Carlson, Eric, "An Approach for Designing Decision Support Systems," Database, Vol. 10, No 3, Winter 1979, pp. 3-15.

Good, Michael D., Whiteside, John A., Wixon, Dennis R., and Jones, Sandra J., "Building a User-Derived Interface," Communications of the Association for Computing Machinery, Vol. 27, No. 10, October 1984, pp. 1032-1043.

Hayes-Roth, Fredrick, "Knowledge-Based Expert Systems," Computer, Vol. 17, No. 10, October 1984, pp. 263-273.

Hayes-Roth, Fredrick, "Rule-Based Systems," Communications of the Association for Computing Machinery, Vol. 28, No. 9, September 1985 pp. 921-932.

Keen, P. G. and Wagner, G. R., "DSS: An Executive Mind-Support System," Datamation, Vol. 25, No. 12, November 1979, pp. 117-122.

Michaelson, Robert and Michie, Donald, "Expert Systems in Business," Datamation, Vol. 29, No. 11, November 1983, pp. 240-246.

Morland, D. Verne, "Human Factors Guidelines for Terminal Interface Design," Communications of the Association for Computing Machinery, Vol. 26, No. 7, July 1983, pp. 484-494.

Nau, Dana S., "Expert Computer Systems," IEEE Computer, Vol. 16, No. 2, February 1983, pp. 63-85.

Slagle, James R. and Hamburger, Henry, "An Expert System for a Resource Allocation Problem," Communications of the Association for Computing Machinery, Vol. 28, No. 9, September 1985 pp. 994-1004.

Stohr, Edward A., and White, Norman H., "User Interfaces for Decision Support Systems: An Overview," International Journal of Policy Analysis and Information Systems, Vol. 6, No. 4, pp. 393-423.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library (Code 0142) Naval Postgraduate School Monterey, CA 93943-5002	2
3. Computer Technology Curriculum Officer (Code 37) Naval Postgraduate School Monterey, CA 93943-5000	1
4. Barry A. Frew (Code 54Fw) Naval Postgraduate School Monterey, CA 93943-5000	2
5. Commanding Officer Naval Supply Center (Code 60) Attn: LCDR Robert L. Beard III Charleston, SC 29408-5000	2
6. Commander Naval Supply Systems Command Attn: LCDR Ted Case (SUP 043A) Washington, DC 20376-5000	1
7. Commanding Officer Navy Fleet Material Support Office Attn: LCDR Winston Buckley P.O. Box 2010 Mechanicsburg, PA 17055-0787	1

END

DTIC

7-86